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


*Department of Computer Engineering, Faculty of Electronics, Wroclaw University of Science and Technology, Wroclaw, 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, Wroclaw University of Science and Technology, Wroclaw, Poland (e-mail: jaroslav.pempera@pwr.edu.pl).
****Institute of Mechanical Engineering, University of Zielona Góra, Zielona Góra, Poland (e-mail: j.patalas@izzp.uiz.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 existence and 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.](image)

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

---

**Preprints of the 21st IFAC World Congress (Virtual)**

**Berlin, Germany, July 12-17, 2020**

**Fig. 1. A transportation system with repetitive tasks.**
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.

![Diagram of a basic system of two cyclic processes (2-P).](image)

- \( P_i \) - 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
- \( O_i \) - output
- \( O_j \) - output

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.
A system of cyclic processes $\text{CPS} = \{P_1, \ldots, P_n, \ldots, P_k\}$, $(i=1,2,\ldots,n)$ is considered. Each process $P_i$ is corresponding to one repetitive transportation task (Fig.1). The process follows a sequence of operations $ZO_i = (O_{i1}, \ldots, O_{ik})$ periodically using different resources defined by $Z_i = (R_{i1}, \ldots, R_{ik})$, 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 $ZO_i$). $OP = \{(O_{ij} | j=1,\ldots,k)\} OP(i) = \{O_{ij}, \ldots, O_{ik}\}$ is a set of the operations $(t_0 - a$ number of the operations), $R_{ij} \in RE$, and $RE = \{R_1, \ldots, R_k\}$ 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 $RES = \{O_{ij}, \ldots, O_{ik}\}$ is a set of resources shared by the processes $(ls - \text{the number of shared resources}), REU$ is a set of unshared resources. The sequence $ZT_i = (r_{i1}, \ldots, r_{i(k)})$ denotes operations times and a cycle time of process $P_i$ is equal to $c_i = \sum_{j=1,\ldots,k} (r_{ij})$, 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 $\text{CPS} = \{P_1, P_2, P_3\}$ 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: $Z_1 = (R_{11},R_{12},R_{13},R_{14}) = (R_1,R_2,R_3,R_4), Z_2 = (R_{21},R_{22},R_{23},R_{24}) = (R_5,R_6,R_7,R_8), Z_3 = (R_{31},R_{32},R_{33},R_{34}) = (R_9,R_4,R_5,R_8)$.

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: $(R_a,R_b) \in E \iff (\exists i \in \{1,\ldots,n\})$
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.5). In the case of the system shown in Fig.4, we have the following 2-process subsystems: 2-PE1(13) - the subsystem of processes SP1 = {P1,P2} sharing R1, and 2-PE2(12) - the processes SP2 = {P2,P3} sharing R2, and 2-PE1(23) - the processes SP3 = {P3,P4} sharing R3 (where indices i(k) of the subsystem 2-PE(k) denote: i - shared resource number; k - numbers of processes sharing resource Rk, and SP1 - a set of processes sharing R1).

There are cyclic processes in the CPS (Fig.4) that use more than one shared resource (e.g. P1 uses R1 and R2, Fig.4), i.e. resources used in different 2-PE subsystems. Then, in the CPS process system, the following situation may occur: the P1 process, which uses the R1 resource and requires R2 resource for the next operation, and the P2 process, which uses the R2 resource and requires R1 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. P1 blocks R1 and P2 blocks R2), 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=\{P1, ..., Pn\}, which is structurally deadlock-free, is analysed as a composition of 2-PE = \{P1,P2\}, subsystems where P1,P2\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=R1=R2=RES is used by two processes \{P1,P2\}, where Z1 = (R1, ..., R2, ..., Rn), and Z2 = (R1r, ..., R2r, ..., Rnr), will be determined on the basis of appropriate conditions developed for the 2-PE 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_1,P_2\} is a composition of two cyclic processes sharing a resource R (Fig.2). The processes P1 and P2 execute periodically sequences Z1, Z2 of the operations using shared resources O1, O2, 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_j = r_i + o_i. Let x(k), x(k)\in N\{0\}, k,l=0,1,2,\ldots,\infty denote the times of starting operations using the shared resource R in subsequent iterations of the processes P1, P2. We assume that 0 \leq x(0) and 0 \leq x(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-PE 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(k) = x(0) + k \cdot C_j and x(k) = x(0) + l \cdot C_j (Fig.5).

Let t_k(\ell)\in N\{0\} define a local start times t_k(\ell)\in [0,c) (1) of process P_\ell, calculated in relation to resource allocation times x(k) of P_\ell, where k,l=0,1,2,\ldots,\infty. The process P_\ell starts an operation with the resource R at the moment x(\ell) = x(k) + t_\ell(\ell), such that x(\ell)\in \{x(k), x(k)+c\}, and the following start times are defined by x'_{\ell}(\ell+1) = x(k) + 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_1,P_2\}, sharing the resource R, process P1 may start to use shared resource only at times x(\ell) + t_\ell(\ell), such that local times t_\ell(\ell)\in [0,c) can be expressed by a relationship:

\begin{equation}
t_\ell(\ell) = f_{\ell j} \cdot D_j + y_{\ell j}
\end{equation}

where: D_j = D_j = \gcd(c_j,c_j) and c_j = D_j m_j_i and c_j = D_j m_j_i & & \& \gcd(m_j_i,m_j_i) = 1 \land m_j_i, \land m_j_i e N \land f_j \in \{0,1, \ldots, m_j_i - 1\} & & \& y_{\ell j} \in \{r_i, D_j - r_j\} & \& \gcd - the greatest common divisor.

The constraint (1) ensures that resource requests x(\ell) of process P_\ell, concerning the shared resource, may occur only within some intervals [x(k)+r_i, x(k)+c_j-r_i], such that r_i \leq t_\ell(\ell) \leq c_j - r_j, which guarantees that no processes wait for the resource.
It can be shown (Wójcik, 2001, 2018) that if the relation (1) holds then also formula describing local start times \( t_j(k) \in [0, c_j) \) of the process \( P_j \) at time intervals \( x(k) \in [x(l), x(l) + c_j) \), and \( x(k) = x_j(l) + t_j(k) \), holds:

\[
 t_j(k) = f_{ij} - D_{ij} + y_j
\]

(2)

where: \( D_{ij} = \text{gcd}(c_i, c_j) \) & \( c_i = D_{ij} \cdot n_{ij} \) & \( c_j = D_{ij} \cdot m_{ij} \) & \( f_{ij} \in [0, 1, \ldots, m_{ij} - 1] \) & \( y_j \in [r_j, D_{ij} \cdot r_j] \) & \( y_j = D_{ij} \cdot y_j \).

The values \( y_j(1) \) and \( y_j(2) \) exist if the intervals \([r_j, D_{ij} - r_j]\) and \([r_j, D_{ij} - r_j]\) are not empty, i.e. when \( r_j \leq D_{ij} - r_j \) and \( r_j \leq D_{ij} - r_j \), 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:

\[
r_j + r_j \leq D_{ij} \quad \& \quad r_j, r_j \in N \quad (3)
\]

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 \( dX_{ij} = x(l) - x(k) \) between any start times \( 0 \leq x(k) \leq x(l) \) is such that the following relations hold:

if \((x(k) \leq x(l) \) then \( y_j = dX_{ij} \mod D_{ij} \) \( \& \ y_j \in [r_j, D_{ij} - r_j], \)

(4)

where \( \mod \) - modulo operation.

The last condition is due to the fact (1), and \( x(l) + w = x(l) + w \cdot c_j = x(k) + t_j(k) + w \cdot c_j \), where \( w \in \mathbb{N} \cup \{0\} \), and the relation \( dX_{ij} = x_j(l) + w \cdot c_j = (x(l) + w \cdot c_j) = f_{ij} \cdot D_{ij} + y_j + w \cdot D_{ij} \cdot m_{ij} = f_{ij} \cdot D_{ij} + y_j \cdot D_{ij} \cdot m_{ij}, \) hence \( dX_{ij} \mod D_{ij} = y_{ij} \).

Similarly, a time distance \( dX_{ij} = x(k) - x(l) \) between any start times \( 0 \leq x(l) \leq x(k) \) is such that the following holds:

if \((x(k) \leq x(l) \) then \( y_{ij} = dX_{ij} \mod D_{ij} \) \( \& \ y_{ij} \in [r_j, D_{ij} - r_j] \)

(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 \( O_a \) operation in the \( P_i \) process, and by \( O_b \) in the \( P_j \) process, i.e. the following dependences are fulfilled \( R(O_a) = R_{a} = R \) and \( R(O_b) = R_{b} = R \), and \( Z_i = (R_{i1}, \ldots, R_{i\alpha}, \ldots, R_{i\alpha}), \) and \( Z_j = (R_{j1}, \ldots, R_{j\beta}, \ldots, R_{j\beta}). \) Suppose that the starting times for operations in subsequent process cycles are defined by \( x_{ia}(k), x_{ib}(k) \in \mathbb{N} \cup \{0\}, \) \( k = 0, 1, 2, \ldots, \infty. \) Let the starting times of the operations \( O_a \) and \( O_b \) at the initial state of the CPS are known, i.e. \( x_{ia}(0) = x_{ja}(0) \) and \( x_{ib}(0) = x_{jb}(0) \), and \( x_{ja}(0), x_{jb}(0) \in \mathbb{N} \cup \{0\}. \) 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 comprised of the 2-PE subsystems sharing resources.

**Theorem 1.** Let a CPS = \{\( P_i, \ldots, P_n \)\} be given. Assume, that the start times of the processes operations are equal to \( x_{ia}(0) = (x_{ia}(0), \ldots, x_{ia}(0)), \) \( \) where \( x_{ia}(0) \in [0, c_i), \) \( x_{ib}(0) = x_{ib}(0) + r_{ja} \), \( \ldots, x_{ia}(0) = x_{ia}(0) + r_{ja} \), \( \) 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 \( R_{a} = R_{b} = R \), and \( R(O_a) = R_{a} = R, \) and \( R(O_b) = R_{b} = R, \) and starting times \( x_{ia}(0) = x_{ib}(0) + \sum_{i = 1}^{\alpha} r_{ja} \) & \( x_{ib}(0) = x_{ia}(0) + \sum_{i = 1}^{\alpha} r_{ja} \) are calculated taking into account \( x_{ia}(0) = [0, c_i) \) and \( x_{ib}(0) = [0, c_i) \). The necessary and sufficient condition for existence of a no-wait cyclic schedule for the CPS with the initial state given by \( x(0) = (x_{ia}(0), x_{ib}(0), \) \( \ldots, x_{ia}(0) \) \( ) \) and \( x_{ia}(0) = [0, c_i) \) is defined as follows:

- for each subsystem 2PE, such that \( 0 \leq x_{ia}(0) \leq x_{ia}(0), \)
  \( \) and \( x_{ia}(0) = x_{ib}(0), \) \( \) and \( x_{ib}(0) = x_{ia}(0), \) \( \) the relation (4) holds;

- for each subsystem 2PE, such that \( 0 \leq x_{ia}(0) \leq x_{ia}(0), \)
  \( \) and \( x_{ia}(0) = x_{ib}(0), \) \( \) and \( x_{ib}(0) = x_{ia}(0), \) \( \) 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 a schedule exists. One may notice that a cycle time of the no-wait schedule is equal to \( T = \text{lcm}(c_1, \ldots, c_n) \), where \( \text{lcm} \) denotes the least common multiple, and \( c_i \) - cycle time of process \( P_i \).

### 3.2 Procedure of no-wait schedule determination

Consider a CPS = \{\( P_1, \ldots, P_n \)\} system with fixed parameters and a fixed initial state \( x(0) = (x_{ia}(0), x_{ib}(0), \ldots, x_{ia}(0)) \), where \( x_{ia}(0) \in [0, c_i) \), 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 \( O_{ib} \) using this resource, i.e. those for which \( R(O_{ia}) = R_{a} = R, \) and \( R(O_{ib}) = R_{b} = R. \)

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 \( D_{ij} = \text{gcd}(c_i, c_j) \). 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 \( x_{ia}(0) = x_{ia}(0) + \sum_{i = 1}^{\alpha} r_{ja} \) & \( x_{ib}(0) = x_{ib}(0) + \sum_{i = 1}^{\alpha} r_{ja} \), where \( x_{ia}(0) \in [0, c_i) \) and \( x_{ib}(0) \in [0, c_i) \).   

4) For starting times such that \( 0 \leq x_{ia}(0) \leq x_{ia}(0), \) calculate \( y_{ij} = dX_{ij} \mod D_{ij}, \) where \( dX_{ij} = x_{ia}(0) - x_{ia}(0). \) If \( y_{ij} \in [r_j, D_{ij} - r_j] \),
then there is no-waiting for the shared resource $R$ in the subsystem $2$-PE. In the same way, for starting times such that $0 \leq x_3(0) \leq x_2(0)$, calculate (5) $y_{ij} = dX_{ij} \mod D_{ij}$, where $dX_{ij} = x_4(0) - x_4(0)$. If $y_{ij} \in [r_j, D_j - r_j]$, 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 $x(0) = (x_1(0), \ldots, x_6(0))$, where $x_i(0) \in [0, c_i)$, 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: $Z_d(1) = (O_1, O_2, 0, 0, 0)$; $Z_{d+1}(1) = (R_1, R_2, R_3, R_4)$; $Z_d(2) = (O_2, O_2, 0, 0, 0)$; $Z_{d+1}(2) = (R_2, R_2, R_2, R_3)$; and $Z_d(3) = (O_3, 0, O_3, 0, 0)$. The set of shared resources $RES = \{R_1, R_2, R_3\}$. The operation times: $Z_{T1} = (r_1, r_2, r_3)$; $Z_{T2} = (r_2, r_3, r_4, r_4) = (1, 1, 3, 1); c_2 = 6$. $Z_{T1} = (r_2, r_3, r_4, r_4) = (1, 1, 3, 1); c_2 = 6$. According to (1): $D_{d+1} = \text{gcd}(c_1, c_2) = 3$, $m_1 = 3$, $m_2 = 2$, and $c_1 = D_1 m_1$, $c_2 = D_2 m_2$, $c_3 = D_3 m_3$, $D_1 = \text{gcd}(c_1, c_2) = 3$, $m_1 = 3$, $m_1 = 4$, and $c_1 = D_1 m_1$, $c_2 = D_2 m_2$, $c_3 = D_3 m_3$. CPS is decomposed into three $2$-process subsystems that are sharing one resource: $2$-PE(1), $2$-PE(2), and $2$-PE(3), the processes $SP_1 = \{P_1, P_2\}$ sharing resource $R_1$, and $2$-PE(2), the processes $SP_2 = \{P_1, P_2\}$ sharing resource $R_2$, and $2$-PE(3), the processes $SP_3 = \{P_1, P_3\}$ sharing $R_1$ for $(k, l)$ i.e. a number of a shared resource, and $(k, l)$ the numbers of processes. The following relations hold: $R(1) = R(O_1) = R_1$; $R(2) = R(O_2) = R_2$; $R(3) = R(O_3) = R_3$; and for $SP_1$, we have $r_1 + r_3 = 2 + 1 = 3 \leq D_{d+1}$; for $SP_2$, we have $r_1 + r_2 = 2 + 1 = 3 \leq D_{d+1}$; and for $SP_3$, we have $r_2 + r_3 = 2 + 1 = 4 \leq D_{d+1}$. 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 $x(0) = (x_1(0), x_2(0), x_3(0))$, where $x_i(0) \in [0, c_i)$, $x_1(0) \in [0, c_1)$, $x_2(0) \in [0, c_2)$, $x_3(0) \in [0, c_3)$, and using it we calculate all starting times of the operations: $x_i(0), x_i(0), x_4(0), x_5(0), x_2(0), x_3(0), x_2(0), x_2(0), x_3(0), x_4(0), x_2(0)$, and $x_3(0), x_2(0), x_3(0), x_4(0)$. For the set $SP_1 = \{P_1, P_2\}$ we calculate the time distance (4), (5) between start times $x_i(0)$, $x_i(0)$ of the operations using the resource $R_1$, i.e. $dX_{ij} = (x_i(0) - x_i(0))$, and $dX_{ij} = (x_i(0) - x_i(0))$, where $dX_{ij}$ denotes the time distance between operations of the processes $P_i, P_j$ 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): if $(x_1(0) \leq x_3(0))$ then $\{y_{ij} = dX_{ij} \mod D_{ij} \land y_{ij} \in [r_j, D_j - r_j]\}$, otherwise if $(x_1(0) > x_3(0))$ then $\{y_{ij} = dX_{ij} \mod D_{ij} \land y_{ij} \in [r_j, D_j - r_j]\}$. Similarly, we define the no-wait constraints for the subsystem $2$-PE(2): if $(x_2(0) \leq x_2(0))$ then $\{y_{ij} = dX_{ij} \mod D_{ij} \land y_{ij} \in [r_j, D_j - r_j]\}$, otherwise if $(x_2(0) > x_3(0))$ then $\{y_{ij} = dX_{ij} \mod D_{ij} \land y_{ij} \in [r_j, D_j - r_j]\}$, and for the subsystem $2$-PE(3): if $(x_3(0) \leq x_3(0))$ then $\{y_{ij} = dX_{ij} \mod D_{ij} \land y_{ij} \in [r_j, D_j - r_j]\}$, otherwise if $(x_3(0) > x_3(0))$ then $\{y_{ij} = dX_{ij} \mod D_{ij} \land y_{ij} \in [r_j, D_j - r_j]\}$. Performing calculations using IBM ILOG CPLEX or just Microsoft Excel tool, in the present case, there are $c_1 = 2, c_2 = 2, c_3 = 2$, and $c_1 = 2, c_2 = 2, c_3 = 2$. 208 different initial states $x(0) = (x_1(0), x_2(0), x_3(0))$, such that $x_1(0) \in [0, 9]$, $x_2(0) \in [0, 6]$, $x_3(0) \in [0, 12]$, and $(x_1(0), x_2(0))$ or $(x_3(0), 0)$, 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 $y(13) = 2, y(12) = 2$, and $y(3) = 4$. A Gantt chart in Fig.6 presents a no-wait cyclic schedule obtained for the initial state $x(0) = (x_1(0) = 1, x_2(0) = 0, x_3(0) = 0)$. Its cycle time: $T = \text{lcm}(9, 6, 12) = 36$. 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 ($R_1, R_2, R_3, R_4$) which are necessary to carry out subsequent operations).

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 calculations 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


