Cyclic processes, cyclic scheduling, timetabling, periodicity, periodic timetables, dispatching rules

Grzegorz BOCEWICZ[*], Zbigniew BANASZAK[**]

CYCLIC PROCESSES SCHEDULING

Abstract

In everyday practice cyclic scheduling problems, especially timetabling ones arise in different application and service domains, e.g., class, train, crew timetabling, and so on. In many cases, e.g., caused by assumed slot size, imposing integer domain results in Diophantine character of problems considered. In that context some classes of cyclic scheduling problems can be seen as non-decidable (undecidable) ones. That means, since system constraints (i.e., parameter domains) determine its behavior (e.g., the space of feasible schedules), hence both system structure configuration and desired schedule have to be considered simultaneously. So, cyclic scheduling problem solution requires that the system structure configuration must be determined for the purpose of processes scheduling, yet scheduling must be done to devise the system configuration. In that context, this contribution provides discussion of some solubility issues concerning cyclic processes dispatching problems.

1. INTRODUCTION

The A cyclic scheduling problem arise in case when system common shared resources have to be allocated to operations in the system composed of concurrently executed cyclic processes as to meet requirements assumed. Operations in cyclic processes are executed along sequences that repeat an indefinite number of times. The scheduling problems considered belong to the class of NP-hard ones and are usually formulated in terms of decision problems, i.e. as searching for an answer whether a solution possessing the assumed features exists or not [10].

In everyday practice they arise in different application domains (such as manufacturing, time-sharing of processors in embedded systems, digital signal processing, and in compilers for scheduling loop operations for parallel or pipelined architectures) as well as service domains (covering such areas as workforce scheduling (e.g., shift scheduling, crew scheduling), timetabling (e.g., train timetabling, aircraft routing and scheduling), and reservations (e.g., reservations with or without slack, assigning classes to rooms) [4, 5, 7, 14, 15]. Such systems belong to a class of so called systems of concurrently flowing cyclic processes (SCCP) [4].

[*] Ph.D. Grzegorz Bocewicz, Department of Computer Science and Management, Koszalin University of Technology, ul. Śniadeckich 2, 75-354 Koszalin, Poland, bocewicz@ie.tu.koszalin.pl

[**] Prof. Zbigniew Banaszak, Department of Business Informatics, Warsaw University of Technology, ul. Narbutta 85, 02-524 Warszawa, Poland, Z.Banaszak@wz.pw.edu.pl

Subway or train traffic can be considered as an example of such kind of systems. Most systems operate several routes, and distinguish them by numbering, names and colors. Some lines may share track with each other, or operate solely on their own right-of-way. Assumption the subway trains following particular metro lines (due to the subway map, that can be seen as a kind of the transit map) can be treated as cyclic processes passing, due to a given timetable, the sequence of stations, allows one to state a question concerning a minimization of the total passenger travel time.

So, if passengers travel between two distinguished locations in the transportation network for which no direct connection exists, i.e., transfers become inevitable, the relevant scheduling problem can be stated in the following way. Given a set of metro lines, each one treated as a repeating sequence of stations. Some lines may share the common stations. Given a *headway time* (interval between the trains), i.e., the fixed interval between the trips of a line sometimes called the *period time*. The question considered is: What is a transportation route between to designated terminal stations in the transportation network providing the shortest travel time subject to above mentioned constraints?

Above stated rather very rough problem formulation can be specified in much more detailed manner, taking into account other constraints (e.g. determining processes interaction – conducted due to rendezvous or mutual exclusion protocol) as well as goal functions (objectives) (e.g. aimed at minimizing dwell and transfer times, number of vehicles, and so on).

Other examples of systems composed of concurrently flowing cyclic processes follow from traffic route organization, e.g. employing the "green wave" concept, as well as railway and airplane traffics. In the case of traffic light control the problem reduces itself to the question: Does there exist such control guaranteeing each route in each direction provides "green wave" vehicles flow? In some sense an alternative (i.e. reverse) question is: Does there exist such traffic light controls guaranteeing some places on an urban traffic route map can be linked by "green waves"? Such solution seems to be of great importance in cases caused by sport or show as well as morning or afternoon rush hour management.

Let us focus, however on problems belonging to the class of non decidable ones. In order to illustrate this fact let us consider the transportation route linking two subway's stations while composed of the three metro lines a, b, and c. Let us assume the route, i.e., the sequence (x, y, z) is specified by number of stations n_i, $i \in \{a, b, c\}$, and time intervals t_i, $i \in \{a, b, c\}$, between two subsequent trains. That means the trains in the line a are scheduled with $t_a$minutes periodicity. The only one train is allowed to run between two subsequent stations. Assume $n_a = 3$, $n_b = 4$, and $n_c = 2$ (in numbers), while $t_a = 1$, $t_b = 2$, and $t_c = 3$ (in minutes). So, the time required for travel between distinguished two stations can be easily calculated as the following sum: $n_a \cdot t_a + (n_b + 1) \cdot t_b + (n_c + 1) \cdot t_c$, end equals to $3 + 9 + 7 = 19$. However in case of opposite the question: what is the route composed of three lines with arbitrarily assumed time intervals t_a, t_b, t_c, guaranteeing the travel time between two stations equals to $T \in N$?, is not so easy.

It should be noted that corresponding equation: $n_a \cdot t_a + (n_b + 1) \cdot t_b + (n_c + 1) \cdot t_c = T$, is a kind of linear Diophantine equation [27, 28] where $n_a, n_b, n_c \in N$ are sought. The problem considered is a decision problem that can be seen as a question stated in some formal system with a yes-or-no answer, depending on the values of some input parameters. Because of its Diophantine character its general case is a non decidable one [12, 13].

Assuming the system's structure imposed by integer domain of variables, i.e., modeled by Diophantine equations determines the system's behavior (seen as a set of feasible schedules), one can formulate the following questions regarding travel scheduling on the base of concurrently flowing cyclic processes framework:

Does the assumed system behavior can be achieved under the given system's structure constraints?

Does there exist the system's structure such that an assumed system behavior can be achieved?

Therefore, taking into account non decidability of Diophantine problems one can easily realize that not all the behaviors are reachable under constraints imposed by system's structure. The similar observation concerns the system's behavior that can be achieved in systems possessing specific structural constraints. To summarize, the conditions guaranteeing solvability of the cyclic processes scheduling are of crucial importance. Their examination may replace exhaustive searching for solution satisfying required system functioning.

Many models and methods have been proposed to solve the cyclic scheduling problem. Among them, the mathematical programming approach (usually IP and MIP), max-plus algebra, constraint logic programming and Petri net frameworks belong to the more frequently used. Most of them are oriented at finding of a minimal cycle or maximal throughput while assuming deadlock-free processes flow. The approaches trying to estimate the cycle time from cyclic processes structure and the synchronization mechanism employed (i.e. rendezvous or mutual exclusion instances) are quite unique. The same regards of the Diophantine nature of timetabling originated cyclic scheduling models.

In that context our main contribution is to propose a new modeling framework enabling to evaluate the cyclic concurrently flowing processes behavior (cyclic steady state) on the base of the given processes topology, dispatching rules employed and an initial state. So, the paper's objective is to provide the observations useful in the course of jobs routing and scheduling in systems composed of cyclic processes interacting each other through mutual exclusion protocol.

The rest of the paper is organized as follows: Section 2 describes the case of manufacturing processes cyclic scheduling. The concept of Diophantine problem is then recalled in Section 3. In Section 4, a case of a multi-mode cyclic manufacturing processes scheduling problem is investigated. Conclusions are presented in Section 5.

2. SYSTEMS OF CONCURRENT CYCLIC PROCESSES

In order to illustrate a system of concurrent cyclic processes let us consider the digraph shown in Fig. 1. The distinguished are three cycles specifying routes of cyclic processes P_1, P_2 and P_3 respectively. Each process route specified by sequence of resources passed on among its execution can interact with other processes through so-called system common resources. So, in case considered the process route are specified as follows:

$$p_1 = (R_6, R_3, R_5), p_2 = (R_2, R_3, R_4), p_3 = (R_1, R_5, R_4), \tag{1}$$

where the resources R_3, R_4, R_5, are shared resources, since each one is used by at least two processes, and the resources R_1, R_2, R_6, are non-shared because each one is exclusively used by only one process. Processes sharing common resources interact each other on the base of mutual exclusion protocol. The possible resources conflicts are resolved with help of assumed priority rules determining the order in which processes make their access to common shared resources (for instance, in case of resource R_4, $\sigma_4 = (P_2, P_3)$ – the priority dispatching rule determines the order in which processes can access to the shared resource R_4, i.e. at first to the process P_2, then to the process P_3, next to P_2 and once again to P_3, and so on).

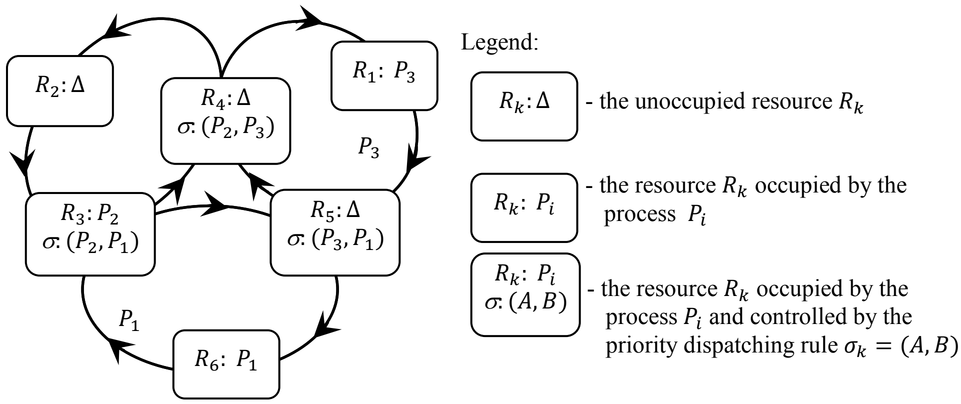


Fig. 1. Process routes structure of SCCP owning three processes

In general case, each process P_i (where $P_i \in P = \{P_1, P_2, \dots, P_n\}$, and n is a number of processes) usually assigned to a production route, executes periodically a sequence of operations using resources defined by a process route $p_i = (R_{j_1}, R_{j_2}, \dots, R_{j_{lr(i)}})$, $j_k \in \{1,2,\dots,m\}$, where $lr(i)$ denotes a length of cyclic process route and m denotes number of resources. In other words $R_{j_k} \in R$, where R is a set of resources of SCCP $R = \{R_1, R_2, \dots, R_m\}$. The time $t_{i,j}$, $t_{i,j} \in \mathbb{N}$, of operation executed on R_j along P_i, is defined in domain of uniform time units ($\mathbb{N}$ – set of natural numbers). The operation times corresponding to resources along P_i create the sequence $T_i = (t_{i,j_1}, t_{i,j_2}, \dots, t_{i,j_{l(i)}})$. To each common shared resource $R_i \in R$ the priority dispatching rule $\sigma_i = (P_{j_1}, P_{j_2}, \dots, P_{j_{lp(i)}})$, $j_k \in \{1,2,\dots,n\}$, $P_{j_k} \in P$ is assigned, where $lp(i) > 1$ determine a number of the i-th dispatching rule σ_i elements.

In that context the system $SC \in SCCP$ can be treated as the following quadruple:

$$SC = (\Pi, T, R, \Theta) \tag{2}$$

where: $\Pi = \{p_1, p_2, \dots, p_n\}$ – the set of process routes,
$T = \{T_1, T_2, \dots, T_n\}$ – the set of sequences of process routes operations times,
$R = \{R_1, R_2, \dots, R_m\}$ – the set of resources,
$\Theta = \{\sigma_1, \sigma_2, \dots, \sigma_m\}$ – the set of dispatching priority rules.

The considered class of $SCCP$ assumes unit operation times for all sequences T_i i.e. $t_{i,j} = 1$.

So, the set of process routes Π determining simple (i.e., local) and repetitively executed cyclic processes can be considered in course of different travel routes prototyping (while linking distinguished system resources), i.e. production flows prototyping. In that context the main question concerns of SCCP cyclic steady state behavior which depends on direction of local process routes as well as on priority rules determining the order in which processes make their access to the common shared resources, and an initial state, i.e. initial process allocation to the system resources. Therefore, assuming such a steady there exists the next question regards of travel time along assumed route linking distinguished resources. Consequently, a SCCP scheduling of plays a pivotal role in multi-product production flow scheduling. So, our assumption is that a way of

SCCP periodicity adjustment can be employed in course of scheduling of production executed in such class of processes.

3. CONCURRENTLY FLOWING CYCLIC PROCESSES SCHEDULING

In the assumed class of systems, different cyclic scheduling problems can be considered. For example, the following questions can be formulated [1, 2]: Does there exists such an initial state that leads to a steady state in which no process waits to access to the common shared resources? What kind of initial states guarantee different steady states under the same sat of priority dispatching rules? What set of priority dispatching rules guarantee, if any, the same rate of resources utilization?

In general case, besides of quantitative features of system behavior the questions may concern the qualitative ones such as deadlock and/or conflict avoidance [8]. For example they may be aimed at satisfaction of conditions, which guarantees system's repetitiveness for a given initial state and/or allocation of dispatching rules.

3.1 Cyclic steady state

Consider the SCCP shown in Fig. 1. Since at each moment system resources are either occupied or not by processes, hence the relevant **processes allocation** can be specified by the following sequence:

$$A = (a_1, a_2, \dots, a_m), \tag{3}$$

where: $a_i \in P \cup \{\Delta\}$, $P = \{P_1, P_2, \dots, P_n\}$ – the set of processes, $a_i = P_k$ means, that the i-th resource R_i is occupied by the k-th process P_k, $a_i = \Delta$ - means, that the i-th resource R_i is unoccupied, m – a number of resources.

For the SCCP from Fig. 1 the processes allocation has the following form:

$$A = (P_3, \Delta, P_2, \Delta, \Delta, P_1), \tag{4}$$

In turn, the current processes allocation is determined by **sequence of semaphores** Z (4) following the order of the priority rules. So, the i-th semaphore z_i, assigned to the i-th resource R_i determines the process allowed to occupy the resource at the moment considered.

$$Z = (z_1, z_2, \dots, z_m), \tag{5}$$

where: $z_i \in P$ - means the name of the process (specified in the i-th dispatching rule σ_i, allocated to the i-th resource) allowed to occupy the i-th resource; for instance $z_i = P_k$ means that at the moment process P_k is allowed to occupy the i-th resource.

For the SCCP from Fig. 1 the sequence of semaphores Z has the following form:

$$Z = (P_3, P_2, P_2, P_2, P_3, P_1). \tag{6}$$

In that context, the following definition of **the $\boldsymbol{k}$-th state** is assumed:

$$S^k = (A^k, Z^k), \tag{7}$$
$$S^k = (({a_1}^k, {a_2}^k, \dots, {a_m}^k), ({z_1}^k, {z_2}^k, \dots, {z_m}^k)). \tag{8}$$

where: A^k - the sequence of processes allocation at the k-th state, Z^k - the sequence of semaphores at the k-th state.

Due to the above definition the state $\boldsymbol{S^k}$ determines the k-th event (at the k-th discrete moment of time) current and subsequent processes alloaction. That means, the assumption concerning the unit operations times $t_{i,j} = 1$, implies execution of each operation corresponds to a system state.

The state $\boldsymbol{S^k}$ determined by (7) and (8), and following the condition (9) is called **a feasible state**:

$$\forall_{i \in \{1,2,\dots,m\}} \, (a_i \in \{z_i, \Delta\}). \tag{9}$$

The set of all feasible states is called a state space $\mathbb{S}$, i.e., $S^k \in \mathbb{S}$.

Consider two subsequent states S^k and S^q:

$$S^k = (({a_1}^k, {a_2}^k, \dots, {a_m}^k), ({z_1}^k, {z_2}^k, \dots, {z_m}^k)), \tag{10}$$
$$S^q = (({a_1}^q, {a_2}^q, \dots, {a_m}^q), ({z_1}^q, {z_2}^q, \dots, {z_m}^q)). \tag{11}$$

Assume the states S^k, S^q are feasible: $S^k, S^q \in \mathbb{S}$. The state $\boldsymbol{S^q}$ **is reachable directly from the state $\boldsymbol{S^k}$** if the following conditions hold:

$$\forall_{\substack{i \in \{1,2,\dots,m\} \\ j \in \{1,2,,,,n\}}} \left[({a_i}^k = \Delta) \wedge \left({a_{\beta_i(P_j)}}^k = {z_i}^k \right) \Rightarrow ({a_i}^{k+1} = {z_i}^k) \right], \tag{12}$$
$$\forall_{\substack{i \in \{1,2,\dots,m\} \\ j \in \{1,2,\dots,n\}}} \left[({a_i}^k = \Delta) \wedge \left({a_{\beta_i(P_j)}}^k \neq {z_i}^k \right) \Rightarrow \left({a_i}^{k+1} \neq P_j \right) \right], \tag{13}$$
$$\forall_{i \in \{1,2,\dots,m\}} [({a_i}^k = \Delta) \Rightarrow ({z_i}^{k+1} = {z_i}^k)], \tag{14}$$
$$\forall_{i \in \{1,2,\dots,m\}} [({a_i}^k \neq \Delta) \wedge ({a_i}^{k+1} \neq \Delta) \Rightarrow [({z_i}^{k+1} = {z_i}^k) \wedge ({a_i}^{k+1} = {a_i}^k)]], \tag{15}$$
$$\forall_{i \in \{1,2,\dots,m\}} [({a_i}^k \neq \Delta) \wedge ({a_i}^{k+1} = \Delta) \Rightarrow ({z_i}^{k+1} = \varphi_i({z_i}^k))], \tag{16}$$
$$\forall_{i \in \{1,2,\dots,m\}} \left[({a_i}^k \neq \Delta) \wedge \left({z_{\alpha_i({a_i}^k)}}^k = {a_i}^k \right) \Rightarrow \left({a_{\alpha_i({a_i}^k)}}^{k+1} = {a_i}^k \right) \wedge ({a_i}^{k+1} = \Delta) \right], \tag{17}$$
$$\forall_{i \in \{1,2,\dots,m\}} \left[({a_i}^k \neq \Delta) \wedge \left({z_{\alpha_i({a_i}^k)}}^k \neq {a_i}^k \right) \Rightarrow ({a_i}^{k+1} = {a_i}^{k+1}) \right], \tag{18}$$

where: m – a number of resources, n – a number of processes,
$\varphi_i(P_j)$ – means the process directly succeeding the process P_j in the i-th priority dispatching rule σ_i, $\varphi_i(P_j) \in P$,
$\beta_i(P_j)$ – means the index of resource directly proceeding the resource R_i, in the j-th process route p_j, $\beta_i(P_j) \in \{1,2, \dots, m\}$,
$\alpha_i(P_j)$ – means the index of resource directly succeeding the resource R_i, in the j-th process route p_j, $\alpha_i(P_j) \in \{1,2, \dots, m\}$.

The assumed set of conditions (12) ÷ (18) is noted B. Feasibility of the state S^q means there is the state S^k such that the state S^q is directly reached from S^k, i.e., the following conditions B, and is denoted by: $S^k \overset{B}{\to} S^q$. In general case, the states S^k and S^q can be linked by other states, e.g. S^r, S^w what leads to the following sequence of transitions: $S^k \overset{B}{\to} S^r \overset{B}{\to} S^w \overset{B}{\to} S^q$, noted in short by $S^k \overset{B,i}{\longrightarrow} S^q$, where: i means the number of states (S^r, S^w) linking states S^k, S^q. So, in case consider $S^k \overset{B,2}{\longrightarrow} S^q$, and $S^k \overset{B}{\to} S^q$ corresponds to $S^k \overset{B,0}{\longrightarrow} S^q$.

The states reachability enables to define a cyclic steady state of SCCP. The finite set of feasible states $Sc = \{S^a, S^b, S^c, \dots, S^d\}$, $Sc \subset \mathbb{S}$ is called **a cyclic steady state** of SCCP if the following condition holds:

$$S^a \overset{B}{\to} S^b \overset{B}{\to} S^c \overset{B}{\to} \dots \overset{B}{\to} S^d \overset{B}{\to} S^a \tag{19}$$

In other words a cyclic steady state means such a set of states in which starting from any distinguished state that is possible to reach the rest of states and finally reach this distinguished state again. Each cyclic steady state is determined by so called period of cyclic steady state Tc. **A cyclic steady state period Tc** is defined in the following way: $Tc = \|Sc\|$. Of course, for any $S^k \in Sc$ the following property holds $S^k \overset{B,(Tc-1)}{\longrightarrow} S^k$.

In that context searching for a cyclic steady state Sc in a given SCCP can be seen as a reachability problem where for an assumed initial state S^0 the state S^k such that holds $S^a \overset{B,i}{\longrightarrow} S^k \overset{B,(Tc-1)}{\longrightarrow} S^k$ is sought.

In order to illustrate the concepts defined let us consider SCCP shown in Fig. 2. For the initial state: $S^0 = (A^0, Z^0)$:

$$A^0 = (P_3, \Delta, P_2, \Delta, \Delta, P_1, P_4),\ Z^0 = (P_3, P_2, P_2, P_2, P_3, P_1, P_4), \tag{20}$$

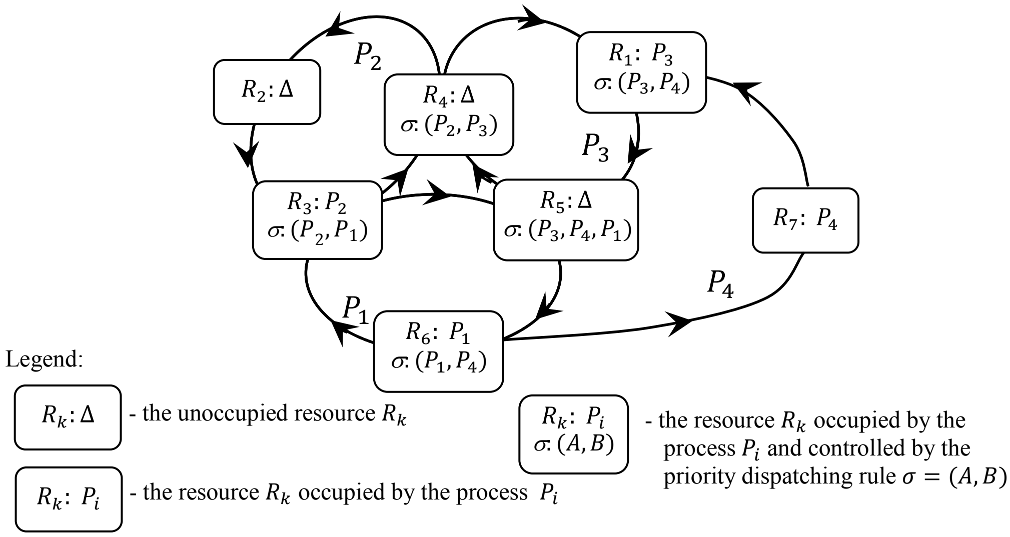


Fig. 2. Process routes structure of SCCP owning four processes

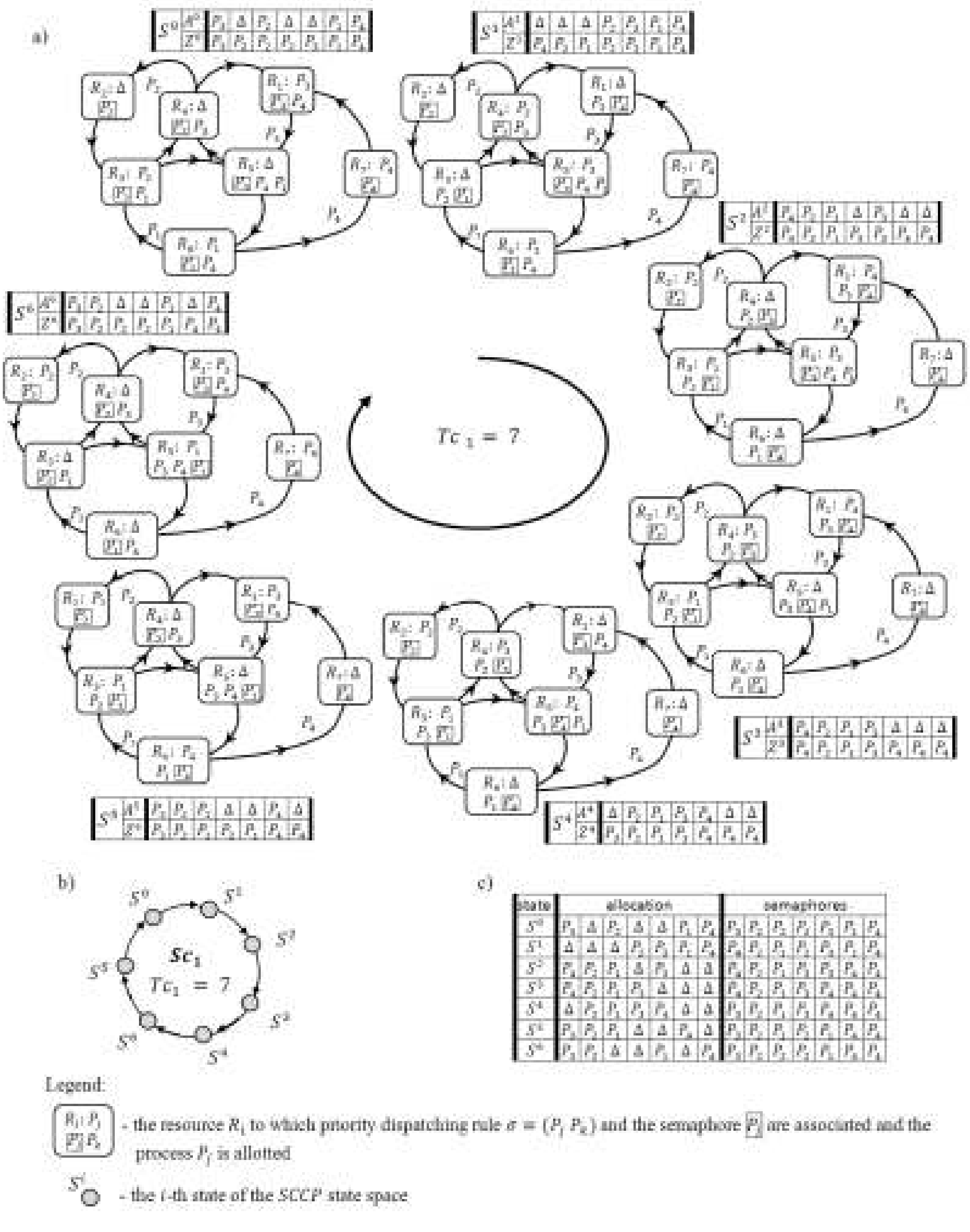


Fig. 3 The SCCP behavior, the case corresponding to the initial state $S^0 = ((P_3, \Delta, P_2, \Delta, \Delta, P_1, P_4), (P_3, P_2, P_2, P_2, P_3, P_1, P_4))$. a) illustration of the transitions between states in the cyclic steady state (the period equal to 7), b) the cyclic steady state of the state space sub-space, c) the states of cyclic steady state

and the following set of priority dispatching rules: $\Theta = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6, \sigma_7\}$:

$$\sigma_1 = (P_3, P_4), \sigma_2 = (P_2), \sigma_3 = (P_2, P_1), \sigma_4 = (P_2, P_3), \sigma_5 = (P_3, P_4, P_1), \quad (21)$$
$$\sigma_6 = (P_1, P_4), \sigma_7 = (P_4).$$

The transient and cyclic steady periods can be seen in Fig. 3. An illustration of possible transitions among the states following condition B and the initial state S^0 (20) is shown in Fig. 3a). The transitions considered are as follows (22):

$$S^0 \xrightarrow{B} S^1 \xrightarrow{B} S^2 \xrightarrow{B} S^3 \xrightarrow{B} S^4 \xrightarrow{B} S^5 \xrightarrow{B} S^6 \xrightarrow{B} S^0 \quad (22)$$

Transition from the state S^0 again to the state S^0 means the SCCP's behavior enables to observe its cyclic manifestation, i.e., cyclic steady period with the cycle length $Tc_1 = 7$. In the case more than one cyclic steady state period may occur, the subsequent ones are distinguished by Sc_i where i means the number of the possible manifestation of SCCP behavior. In each case a cyclic steady state period can be specified by the sequence of states occurring in its execution:

$$Sc_1 = \{S^0, S^1, S^2, S^3, S^4, S^5, S^6\}. \quad (23)$$

For illustration let us consider the Fig. 3b) showing the advanced (more detailed) version from Fig. 3a). Fig. 3b) illustrates a part of state space $\mathbb{S}$ creating the cyclic steady state Sc_1. In turn the table-like representation of the cyclic steady state period Sc przeshow the Fig.3c). For the sake of simplicity the representations used in Fig. 3b) and Fig. 3c) will be used in the rest of considerations.

With any cyclic steady state period of the SCCP the relevant cyclic realization of processes' operations is related. The illustration of particular operations is show in the form of the Gantt's chart, e.g., from Fig. 4. In the case considered the Gantt's chart provides allocation of A^k processes in the k states (assuming the unit operation times the k-th state corresponds to the k-th event occurring in the i-th unit of time). In the Fig.4 three periods of SCCP behavior are distinguished. In the case considered the initial state belong to the cyclic steady state, that means the SCCP starts its functioning without any transient period, however under the same set of priority dispatching rules the relevant SCCP behavior (see the Gant's diagram in Fig 5).

Transition from the state (20) to the sate (24) leads to the new cyclic steady state Sc_2 with periodicy $Tc_2 = 9$. It means the cyclic steady state Sc_2 is reachable from the state (20) through two transition satates.

This observation can be summarized: the different initial states may lead to different steady states under the same set of priority dispatching rules.

The steady state periods shown in Gantt's chart terms in Fig. 4 and Fig. 5 can be also recognized in terms of already introduced state space concept. For the illustration let us consider the part of the state space encompassing the behavior of the SCCP form Fig. 2. This part, which is shown in Fig. 6, consists only the 147 states from the whole state space encountering of 222 states. Some of them as the state S^0 lead to cyclic steady state period length of 7 (i.e. the cyclic steady state Sc_1 from the Fig 3b)), some of them as the state S^7 lead to cyclic steady state period length of 9 (the cyclic steady state Sc_2, see Fig. 5)), someone else as the state S^{18} lead to the deadlock states (the deadlock states are graphically distinguished by the following symbol ⊗). The deadlock state means the state in which at least one process cannot continue its activity.

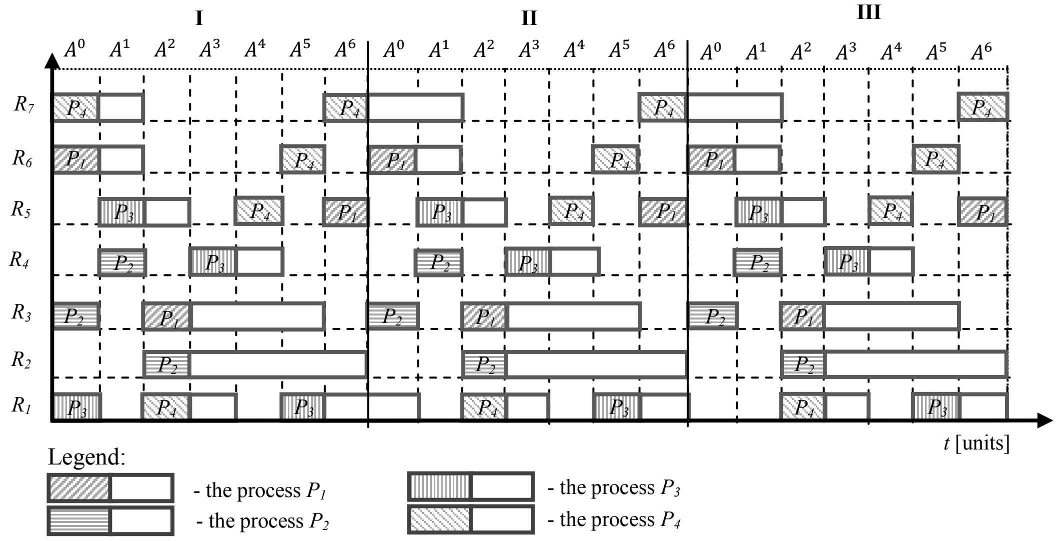


Fig. 4 Gantt's chart of the SCCP behavior. The case corresponding to the initial state $S^0 = ((P_3, \Delta, P_2, \Delta, \Delta, P_1, P_4), (P_3, P_2, P_2, P_2, P_3, P_1, P_4))$.

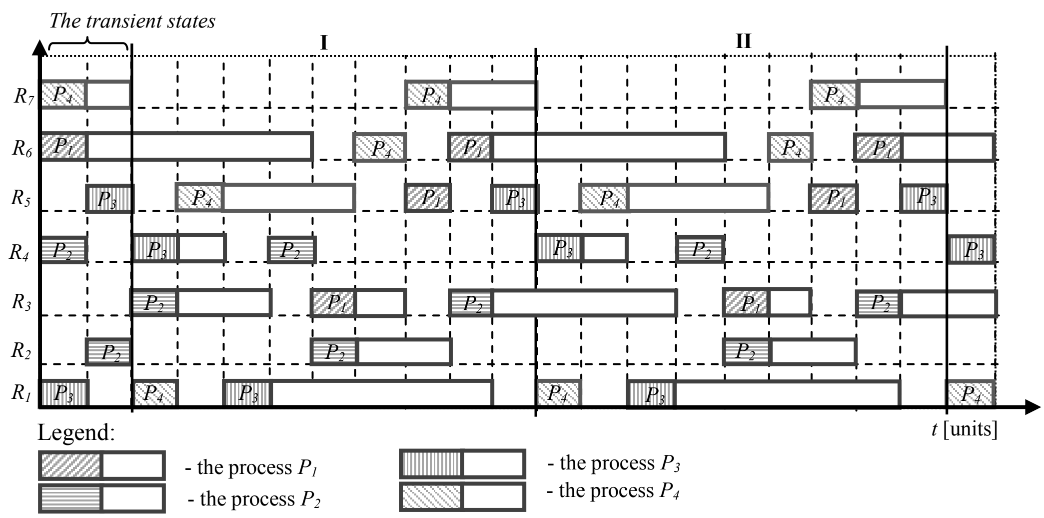
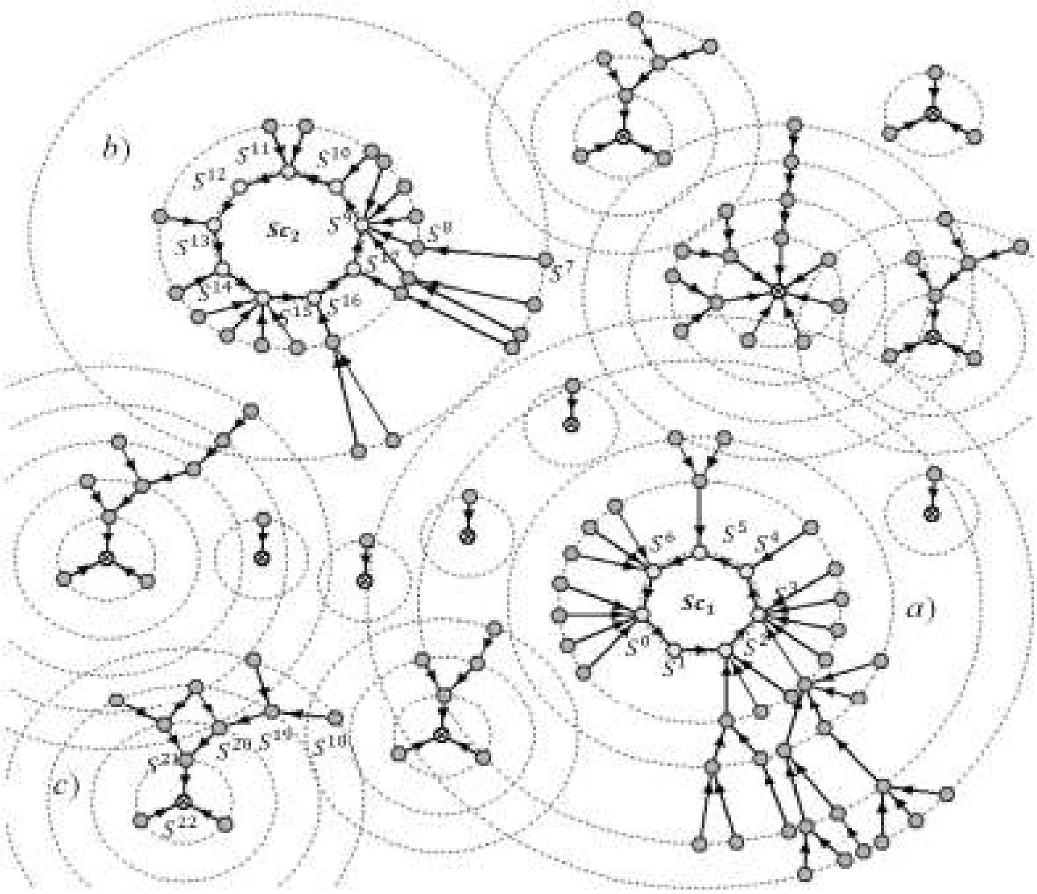


Fig. 5 Gantt's chart of the SCCP behavior. The case corresponding to the initial state $S^0 = ((P_3, \Delta, \Delta, P_2, \Delta, P_1, P_4), (P_3, P_2, P_2, P_2, P_3, P_1, P_4))$

Besides of initial process allocation and priority rules determining the order in which processes make their access to the common shared resources the cyclic steady state behavior depends on direction of local process routes as well.



a') The period $Tc_1 = 7$

state	allocation							semaphores						
S^0	P_3	Δ	P_2	Δ	Δ	P_1	P_4	P_3	P_2	P_1	P_2	P_3	P_1	P_4
S^1	Δ	Δ	Δ	P_2	P_3	P_1	P_4	P_4	P_2	P_1	P_2	P_3	P_1	P_4
S^2	P_4	P_3	P_1	Δ	P_3	Δ	Δ	P_4	P_3	P_1	P_2	P_3	P_4	P_4
S^3	P_4	P_2	P_1	P_3	Δ	Δ	Δ	P_4	P_2	P_1	P_2	P_4	P_3	P_4
S^4	Δ	P_2	P_1	P_3	P_4	Δ	Δ	P_3	P_2	P_1	P_3	P_4	P_4	P_4
S^5	P_3	P_3	P_1	Δ	Δ	P_4	Δ	P_3	P_3	P_1	P_2	P_1	P_4	P_4
S^6	P_3	P_3	Δ	Δ	P_1	Δ	P_4	P_3	P_2	P_2	P_3	P_1	P_4	P_4

b') The period $Tc_2 = 9$

The transient states: S^7, S^8

state	allocation							semaphores						
S^7	P_3	Δ	Δ	P_2	Δ	P_1	P_4	P_3	P_2	P_2	P_2	P_3	P_1	P_4
S^8	Δ	P_2	Δ	Δ	P_3	P_1	P_4	P_4	P_1	P_2	P_1	P_3	P_1	P_4
S^9	P_4	Δ	P_2	P_3	Δ	P_1	Δ	P_4	P_1	P_2	P_3	P_4	P_1	P_4
S^{10}	Δ	Δ	P_2	P_3	P_4	P_1	Δ	P_3	P_2	P_2	P_3	P_4	P_1	P_4
S^{11}	P_3	Δ	P_2	Δ	P_4	P_1	Δ	P_3	P_2	P_2	P_3	P_4	P_1	P_4
S^{12}	P_3	Δ	Δ	P_2	P_4	P_1	Δ	P_3	P_1	P_1	P_2	P_4	P_1	P_4
S^{13}	P_3	P_2	P_1	Δ	P_4	Δ	Δ	P_3	P_1	P_1	P_3	P_4	P_4	P_4
S^{14}	P_3	P_2	P_1	Δ	Δ	P_4	Δ	P_3	P_2	P_1	P_3	P_1	P_4	P_4
S^{15}	P_3	P_2	Δ	Δ	P_1	Δ	P_4	P_3	P_2	P_2	P_3	P_1	P_1	P_4
S^{16}	P_3	Δ	P_2	Δ	Δ	P_1	P_4	P_3	P_2	P_2	P_3	P_3	P_1	P_4
S^{17}	Δ	Δ	P_2	Δ	P_3	P_1	P_4	P_4	P_2	P_2	P_3	P_3	P_1	P_4

c') The deadlock state

state	allocation							semaphores						
S^{18}	P_3	P_2	Δ	Δ	P_1	P_4	Δ	P_3	P_2	P_1	P_2	P_1	P_1	P_4
S^{19}	P_3	P_2	Δ	Δ	P_1	Δ	P_4	P_3	P_2	P_1	P_2	P_1	P_1	P_4
S^{20}	P_3	P_2	Δ	Δ	Δ	P_1	P_4	P_3	P_2	P_1	P_2	P_3	P_1	P_4
S^{21}	Δ	P_2	P_1	Δ	P_3	Δ	P_4	P_4	P_2	P_1	P_2	P_3	P_4	P_4
S^{22}	P_4	P_2	P_1	Δ	P_3	Δ	Δ	P_4	P_2	P_1	P_2	P_3	P_4	P_4

Legend:

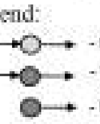


Fig. 6 The state space $\mathbb{S}$ of the SCCP from Fig. 2

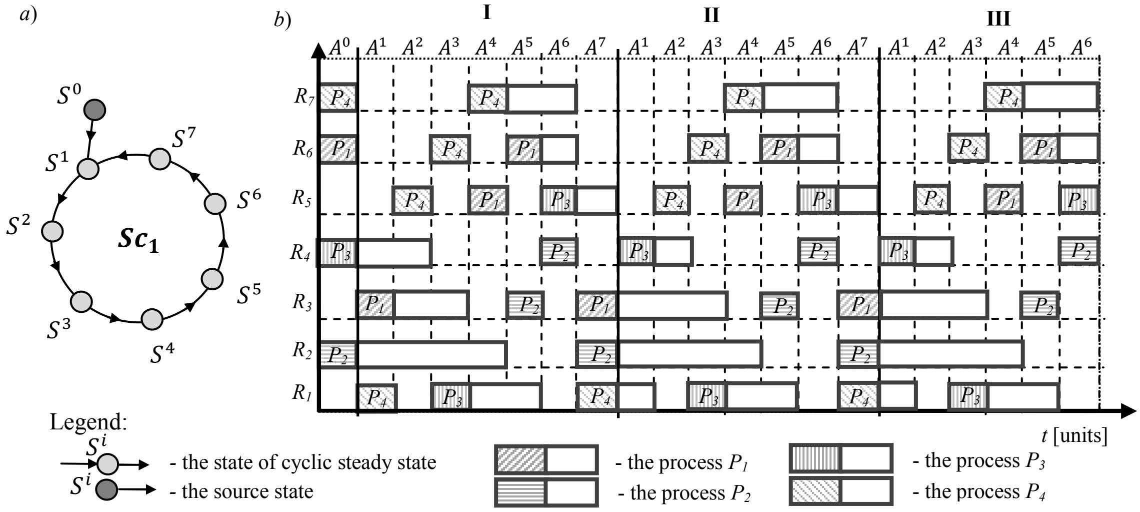


Fig. 7 The SCCP behavior corresponding to the initial state $S^0 = ((\Delta, P_2, \Delta, P_3, \Delta, P_1, P_4), (P_4, P_2, P_1, P_3, P_4, P_1, P_4))$ and dispatching rules $\sigma_1 = (P_4, P_3), \sigma_2 = (P_2), \sigma_3 = (P_1, P_2), \sigma_4 = (P_3, P_2), \sigma_5 = (P_4, P_1, P_3), \sigma_6 = (P_1, P_4), \sigma_7 = (P_4)$, a) the part of the state space $\mathbb{S}$, b) the Gantt's chart

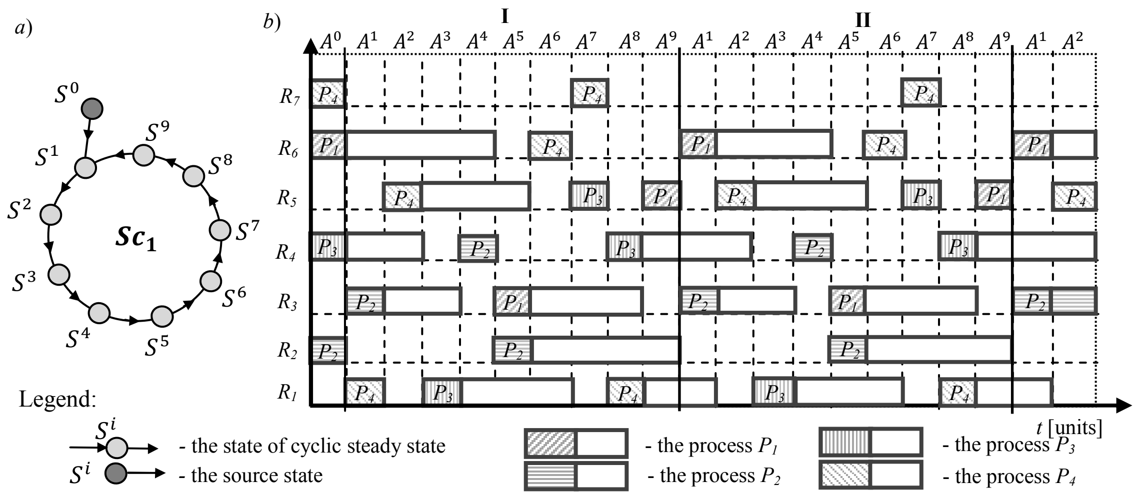


Fig. 8 The SCCP behavior corresponding to the initial state $S^0 = ((\Delta, P_2, \Delta, P_3, \Delta, P_1, P_4), (P_4, P_2, P_1, P_3, P_4, P_1, P_4))$, and dispatching rules: $\sigma_1 = (P_4, P_3), \sigma_2 = (P_2), \sigma_3 = (P_2, P_1), \sigma_4 = (P_3, P_2), \sigma_5 = (P_4, P_3, P_1), \sigma_6 = (P_1, P_4), \sigma_7 = (P_4)$ a) the part of the state space $\mathbb{S}$, b) the Gantt's chart

In case of the following initial state $S^0 = (A^0, Z^0)$:

$$A^0 = (P_3, \Delta, \Delta, P_2, \Delta, P_1, P_4),\ Z^0 = (P_3, P_2, P_2, P_2, P_3, P_1, P_4), \tag{24}$$

The deadlock state can be observed in the SCCP from the Fig. 6c) assuming the following initial state:

$$S^{18} = ((P_3, P_2, \Delta, \Delta, P_1, P_4, \Delta), (P_3, P_2, P_1, P_2, P_1, P_1, P_4)). \tag{25}$$

Trajectories linking above mentioned states with relevant cyclic steady states and the deadlock state are shown in tables Fig. 6a'), b'), and c'), respectively.

In turn, assuming the same initial state the different sets of dispatching rules can lead either to different cyclic steady states or to the deadlock state. For illustration of case the different dispatching rules can lead to different cyclic steady states from the same initial state consider the SCCP in Fig. 7b) and Fig. 8b) the resultant cyclic steady states are characterized with the period $Tc_1 = 7$ and $Tc_2 = 9$ respectively. The graphical illustration of the corresponding states space is shown in Fig. 7a) and Fig. 8a).

In order to summarize the steady state space generated by structure of SCCP, i.e., processes routes topology and sets of priority dispatching rules, as well as possible behaviors following different initial states allow one to classify the systems considered as a kind of discrete event systems [2]. In that context the main question concerns of assumed cyclic steady states reachability. Assuming the cycle length characterizing the steady state period means a number of states determining this cycle the question considered can be stated as follows: what is an initial state and a set of priority dispatching rules, if any, guaranteeing assumed cycle length in a given SCCP? Such question assumes implicitly that periodic behavior of the whole SCCP is limited by lowest common multiple (l.c.m.) of elementary processes cycle lengths, where l.c.m. plays a lower bound of possible cycles describing periodic behavior of the SCCP.

3.2 Diophantine model

In order to illustrate the Diophantine character of the model under study, let us consider the system of concurrently flowing cyclic processes shown in Fig. 1. At the initial state:

$$S^0 = ((P_3, \Delta, P_2, \Delta, \Delta, P_1), (P_3, P_2, P_2, P_2, P_3, P_1)), \tag{26}$$

the cyclic steady state system behavior, illustrated by Gantt's chart (see Fig. 9), is characterized by cycle $Tc = 5$ (obtained under assumption $t_{3,1} = t_{5,1} = t_{6,1} = t_{2,2} = t_{3,2} = t_{4,2} = t_{3,1} = t_{4,3} = t_{5,3} = 1$).

In general case, the periodicities of the cyclic steady states as well as corresponding sets of dispatching rules can be calculated from the linear Diophantine equation of the following form: $3y + 3x = Tc + z$, where x, y, z mean the numbers of processes executions, respectively P_1, P_2, P_3, within the one period of cyclic steady state Sc. The formulae considered have been obtained through the following transformations, following the assumptions below:

- the initial state and set of dispatching rules guarantee that an admissible solution exists (i.e. cyclic steady state),
- the structure of the graph model is consistent.

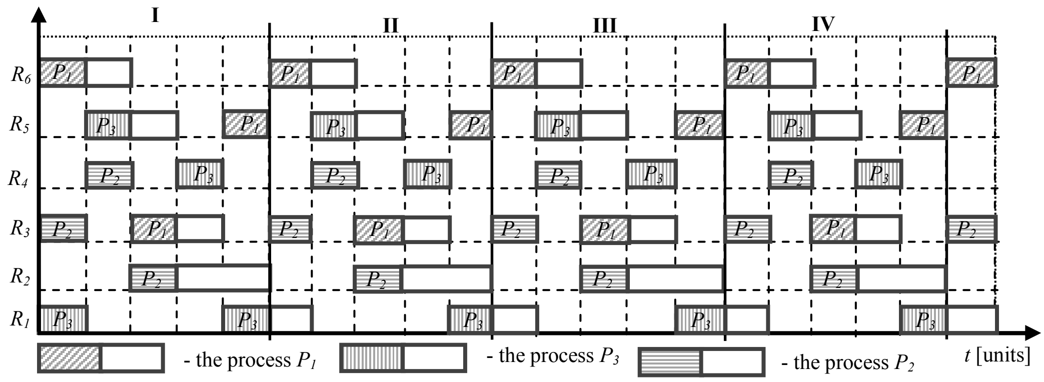


Fig. 9 Gantt's chart of the SCCP (Fig. 1) behavior corresponding to the initial state: $S^0 = ((P_3, \Delta, P_2, \Delta, \Delta, P_1), (P_3, P_2, P_2, P_2, P_3, P_1))$ and set of priority dispatching rulet: $\sigma_1 = (P_3), \sigma_2 = (P_2), \sigma_3 = (P_2, P_1), \sigma_4 = (P_2, P_3),\ \sigma_5 = (P_3, P_1), \sigma_6 = (P_4)$

Consider the following set of equations (27)-(29):

$$x \cdot (t_{3,1} + t_{5,1} + t_{6,1}) + y \cdot t_{3,2} + z \cdot t_{5,3} = Tc, \tag{27}$$
$$y \cdot (t_{2,2} + t_{3,2} + t_{4,2}) + x \cdot t_{3,1} + z \cdot t_{4,3} = Tc, \tag{28}$$
$$z \cdot (t_{1,3} + t_{4,3} + t_{5,3}) + x \cdot t_{5,1} + y \cdot t_{4,2} = Tc, \tag{29}$$

where: $t_{i,j}$ – the execution time of the operations executed on the resource R_i along the process P_j,
Tc – okres przebiegu cyklicznego,
x, y, z – the numbers of processes executions, respectively P_1, P_2, P_3, within the one period of cyclic steady state Sc.

Subtracting equation (29) from equation (28) the resulting equation has the form:

$$y \cdot t_{3,2} + y \cdot t_{2,2} + x \cdot t_{3,1} - z \cdot t_{1,3} - z \cdot t_{5,3} - x \cdot t_{5,1} = 0 \tag{30}$$

and after adding it to equation (27), the resultant formulae has the form:

$$y \cdot (2t_{2,1}\ +\ t_{2,2})\ +\ x \cdot (2t_{2,1}\ +\ t_{1,1}) - z \cdot t_{1,3}\ =\ Tc\,. \tag{31}$$

Consequently, the obtained equation:

$$y \cdot N\ +\ x \cdot M\ =\ Tc\ +\ z \cdot K \tag{32}$$

is Diophantine equation, where $N\ =\ M\ =\ 3$ and $K\ =\ 1$ under the following assumption $t_{3,1} = t_{5,1} = t_{6,1} = t_{2,2} = t_{3,2} = t_{4,2}\ = t_{3,1} = t_{4,3} = t_{5,3} = 1$, i.e. it takes the following form: $3y\ +\ 3x\ =\ Tc\ +\ z$. First three solutions of this equation are shown in Table 1. It should be noted however, that since the assumed set of priority dispatching rules:

$$\sigma_1 = (P_3), \sigma_2 = (P_2), \sigma_3 = (P_2, P_1), \sigma_4 = (P_2, P_3),\ \sigma_5 = (P_3, P_1), \sigma_6 = (P_4), \tag{33}$$

forces alternating access of processes to the shared resources, i.e., $x = y = z$, hence the solution corresponding to the cyclic steady state $Tc = 6$ cannot be reached. Such behavior is not feasible in case of assumed dispatching priority rules.

Tab. 1. First three solutions of the Diophantine equation: $3y + 3x = Tc + z$.

$\boldsymbol{Tc}$	$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{z}$
5	1	1	1
6	1	2	3
10	2	2	2

This might be considered in cases assuming the set of dispatching rules encompassing required frequencies of processes occurrence, e.g.,

$$\sigma_1 = (P_3), \sigma_2 = (P_2), \sigma_3 = (P_2, P_2, P_1), \sigma_4 = (P_3, P_3, P_3, P_2, P_3, P_2),\\ \sigma_5 = (P_1, P_3, P_3, P_3), \sigma_6 = (P_4), \tag{34}$$

The case considered follows from the facts that the Diophantine problems:

have fewer equations than unknown variables and involve finding integers, which satisfy all equations,

can be set up taking into account different structural and behavioral factors, e.g. already mentioned different frequencies of processes occurrence in a priority dispatching rule, different idle times the processes have to spend waiting for their access to shared resources (forced by mutual exclusion protocol following dispatching rules employed).

In order to illustrate the above mentioned cases let us consider once again the SCCP shown in Fig. 1. And let us assume the following frequencies of local processes repetition in the whole SCCP cycle: $x{:}y{:}z = 1{:}2{:}1$. The values considered $x{:}y{:}z$ follows from the assumed blow priority dispatching rules:

$$\sigma_1 = (P_3), \sigma_2 = (P_2), \sigma_3 = (P_2, P_1, P_2), \sigma_4 = (P_2, P_3, P_2),\ \sigma_5 = (P_3, P_1), \sigma_6 = (P_4), \tag{35}$$

The equations are of the following form:

$$\begin{cases} 3x + 2y + z = Tc \\ 3z + 2y + x = Tc \quad x, y, z, Tc \in N \\ 3y + x + z = Tc \end{cases} \tag{36}$$

The solution ($Tc = 8$ for $x = 1$, $y = 2$, $z = 1$) is illustrated in Fig. 10.
In case when $x{:}y{:}z = 3{:}2{:}1$, and

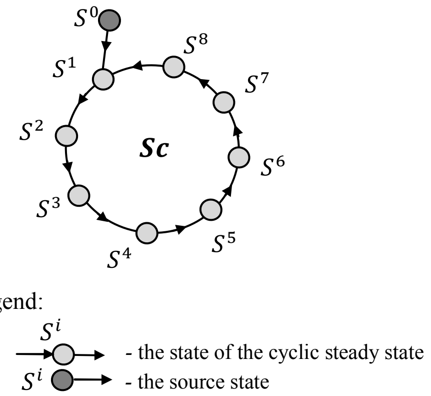
$$\sigma_1 = (P_3), \sigma_2 = (P_2), \sigma_3 = (P_2, P_1, P_2, P_1, P_1), \sigma_4 = (P_2, P_3, P_2),\\ \sigma_5 = (P_3, P_1, P_1, P_1), \sigma_6 = (P_4), \tag{37}$$

i.e. the set of Diophantine equations has the following form:

$$\begin{cases} 3x + 1y + z = Tc \\ \frac{5}{3}x + 3y + z = Tc \\ \frac{7}{3}x + y + 3z = Tc \end{cases} \quad x, y, z, Tc \in N \tag{38}$$

The solution ($Tc = 12$ for $x = 3$, $y = 2$, $z = 1$) is shown in Fig. 11

a)

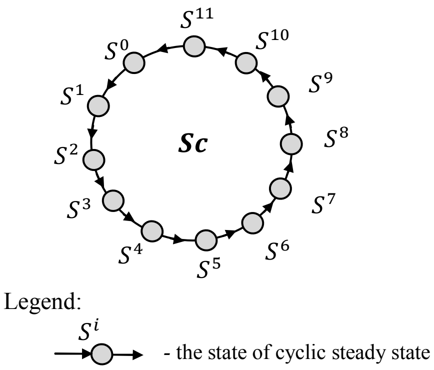


b)

State	allocation						semaphores					
S^0	P_3	Δ	P_2	Δ	Δ	P_1	P_3	P_2	P_2	P_2	P_3	P_1
S^1	Δ	Δ	Δ	P_2	P_3	P_1	P_3	P_2	P_1	P_2	P_3	P_1
S^2	Δ	P_2	P_1	Δ	P_3	Δ	P_3	P_2	P_1	P_3	P_3	P_1
S^3	Δ	P_2	P_1	P_3	Δ	Δ	P_3	P_2	P_1	P_3	P_1	P_1
S^4	P_3	P_2	Δ	Δ	P_1	Δ	P_3	P_2	P_2	P_2	P_1	P_1
S^5	P_3	Δ	P_2	Δ	Δ	P_1	P_3	P_2	P_2	P_2	P_3	P_1
S^6	Δ	Δ	Δ	P_2	P_3	P_1	P_3	P_2	P_2	P_2	P_3	P_1
S^7	Δ	P_2	Δ	Δ	P_3	P_1	P_3	P_2	P_2	P_2	P_3	P_1
S^8	Δ	Δ	P_2	Δ	P_3	P_1	P_3	P_2	P_2	P_2	P_3	P_1

Fig. 10 The SCCP (from Fig. 1) behavior corresponding to the dispatching rules: $\sigma_1 = (P_3), \sigma_2 = (P_2), \sigma_3 = (P_2, P_1, P_2), \sigma_4 = (P_2, P_3, P_2),\ \sigma_5 = (P_3, P_1), \sigma_6 = (P_4)$, a) the part of the state space $\mathbb{S}$, b) the table including the states of the cyclic steady state

a)



b)

State	allocation						semaphores					
S^0	P_3	Δ	P_2	Δ	Δ	P_1	P_3	P_2	P_2	P_2	P_3	P_1
S^1	Δ	Δ	Δ	P_2	P_3	P_1	P_3	P_2	P_1	P_2	P_3	P_1
S^2	Δ	P_2	P_1	Δ	P_3	Δ	P_3	P_2	P_1	P_3	P_3	P_1
S^3	Δ	P_2	P_1	P_3	Δ	Δ	P_3	P_2	P_1	P_3	P_1	P_1
S^4	P_3	P_2	Δ	Δ	P_1	Δ	P_3	P_2	P_2	P_2	P_1	P_1
S^5	P_3	Δ	P_2	Δ	Δ	P_1	P_3	P_2	P_2	P_2	P_1	P_1
S^6	P_3	Δ	Δ	P_2	Δ	P_1	P_3	P_2	P_1	P_2	P_1	P_1
S^7	P_3	P_2	P_1	Δ	Δ	Δ	P_3	P_2	P_1	P_2	P_1	P_1
S^8	P_3	P_2	Δ	Δ	P_1	Δ	P_3	P_2	P_1	P_2	P_1	P_1
S^9	P_3	P_2	Δ	Δ	Δ	P_1	P_3	P_2	P_1	P_2	P_1	P_1
S^{10}	P_3	P_2	P_1	Δ	Δ	Δ	P_3	P_2	P_1	P_2	P_1	P_1
S^{11}	P_3	P_2	Δ	Δ	P_1	Δ	P_3	P_2	P_2	P_2	P_1	P_1

Fig. 11 The SCCP (from Fig. 1) behavior corresponding to the dispatching rules: $\sigma_1 = (P_3), \sigma_2 = (P_2), \sigma_3 = (P_2, P_1, P_2, P_1, P_1), \sigma_4 = (P_2, P_3, P_2),\ \sigma_5 = (P_3, P_1, P_1, P_1), \sigma_6 = (P_4)$, a) the part of the state space $\mathbb{S}$, b the table including the states of the cyclic steady state

Taking into account different frequencies of processes occurrence in a priority dispatching rule and the different idle times the processes have to spend waiting for their access to shared resources one may consider general approach to the cycle time evaluation. Let us assume the cycle time of the whole SCCP from Fig. 12. Follows the formulae: $Tc = lcm(Tp_1, Tp_2, Tp_3)$, where $lcm(k, l, i)$ means the lowest common multiple of $k, l, i \in N$, and Tp_i - is the cycle time of the local process P_i.

Assuming the same frequency $z = y = x = 1$, and the operation times $t_{i,j} = 1$, $j \in \{1, \dots, 7\}$, $i \in \{1, \dots, 3\}$ the following set of Diophantine equations is considered:

$$Tp_1 = t_{1,5} + t_{1,6} + t_{1,7} + \alpha = 3 + \alpha \tag{38}$$

$$Tp_2 = t_{2,1} + t_{2,2} + t_{2,4} + \beta = 3 + \beta \tag{39}$$

$$Tp_3 = t_{3,2} + t_{3,3} + t_{3,5} + \gamma = 3 + \gamma \tag{40}$$

where: α, β, γ - mean the idle times the processes have to spend waiting for their access to shared resources.

Values of $\alpha, \beta, \gamma \in \{0,1,2, \dots\}$ depend on the SCCP's topology. In case considered, see Fig. 12 the process following P_1 may await on the resource R_7. In case of P_2 may await on the resource R_4, and in case of P_3 on resource R_3 and R_5 (corresponding waiting times are: γ_1, γ_2).

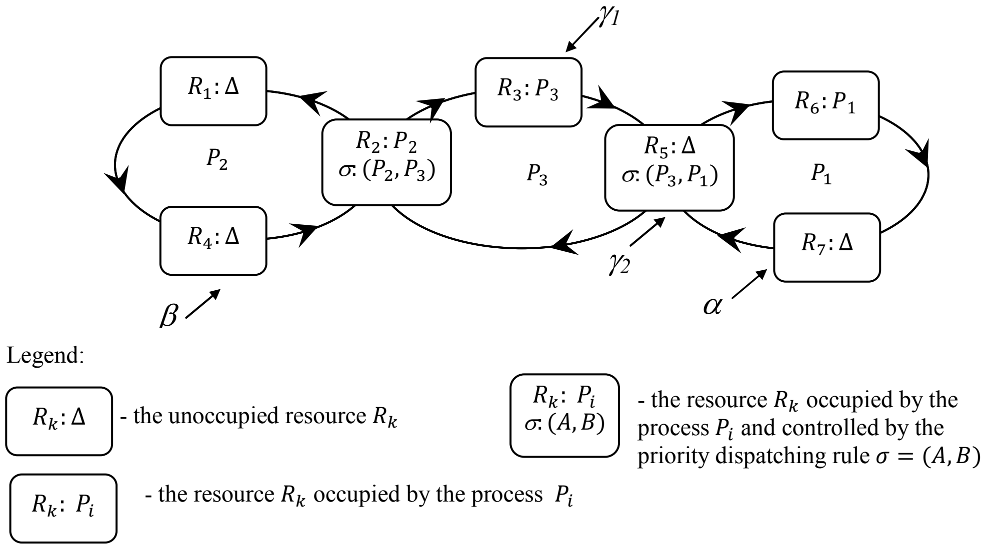


Fig. 12. Process routes structure of SCCP owning three processes

That is easy to note that in the case considered $\alpha, \beta, (\gamma_1 + \gamma_2) \in \{0,1\}$. Consequently,

$$Tc = lcm((3 + \alpha), (3 + \beta), (3 + \gamma)), \tag{41}$$

where: $\gamma = \gamma_1 + \gamma_2$; $\gamma_1, \gamma_2 \in \{0,1\}$; $\alpha \in \{0,1\}$; $\beta \in \{0,1\}$.
Taking into account all the possible substitutions the possible solutions are:

$$Tc \in \{3, 4, 12, 15, 20, 60\}, \tag{42}$$

Since the P_3 can be executed without two waiting i.e., before both R_3 and R_5, hence the set of possible solutions can be as follows: $T \in \{3, 4, 12\}$. Finally, assuming $\gamma_1 + \gamma_2 = 1$, i.e.

$$Tc = lcm((3 + \alpha), (3 + \beta), (3 + \gamma_1 + \gamma_2)), \tag{43}$$

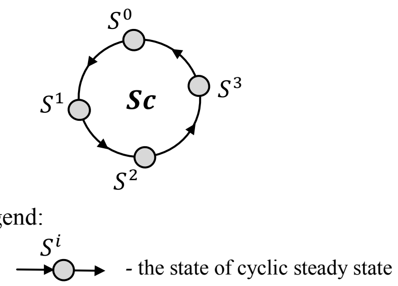
for $\gamma_1, \gamma_2 \in \{0,1\}$, $\alpha \in \{0,1\}$, $\beta \in \{0,1\}$ the cycle time obtained from the formulae below:

$$Tc = lcm((3 + \alpha), (3 + \beta), 4), \tag{44}$$

where: $\alpha \in \{0,1\}$, $\beta \in \{0,1\}$,
equals to $Tc = 4$. The graphical illustration of solution obtained is show in Fig. 13.

In the context considered the following questions belong to the most frequently asked: Are there any solutions? Are there any solutions beyond some that are easily found by inspection? Are there finitely or infinitely many solutions? Can all solutions be found, in theory? Can one compute full list of solutions, i.e. the whole state space?

a)



b)

State	allocation							semaphores						
S^0	Δ	P_2	P_3	Δ	Δ	P_1	Δ	P_2	P_2	P_3	P_2	P_3	P_1	P_1
S^1	P_2	Δ	Δ	Δ	P_3	Δ	P_1	P_2	P_3	P_3	P_2	P_3	P_1	P_1
S^2	Δ	P_3	Δ	P_2	Δ	Δ	P_1	P_2	P_3	P_3	P_2	P_1	P_1	P_1
S^3	Δ	Δ	P_3	P_2	Δ	P_1	Δ	P_2	P_2	P_3	P_2	P_1	P_1	P_1

Fig. 13 The SCCP (from Fig. 12) behavior corresponding to initial state: $S^0 = ((\Delta, P_2, P_3, \Delta, \Delta, P_1, \Delta), (P_2, P_2, P_3, P_2, P_3, P_1, P_1))$, and the dispatching rules: $\sigma_1 = (P_3), \sigma_2 = (P_2), \sigma_3 = (P_2, P_1, P_2, P_1, P_1), \sigma_4 = (P_2, P_3, P_2),$ $\sigma_5 = (P_3, P_1, P_1, P_1), \sigma_6 = (P_4)$, a) the part of the state space $\mathbb{S}$, b) the table including the states of the cyclic steady state

To summarize, the cyclic steady state generation problem has the goal to find a cycle time Tc which fulfils the local processes frequency requirements and is feasible under assumed SCCP structural constraints. More formally, a decision problem (i.e. searching for the initial state and a set of dispatching rules) can be seen as a question stated in some formal system with a yes-or-no answer, depending on the values of some input parameters. The decision problems fall into two categories: decidable and non decidable problems [12, 13]. In this context, the primary question is to what kind of above-mentioned problems the real-life cyclic scheduling ones belong.

A decision problem is called decidable or effectively solvable if an algorithm exists which terminates after a finite amount of time and correctly decides whether or not a given number belongs to the set. A classic example of a decidable decision problem is the set of prime numbers. It is possible to effectively decide whether a given natural number is the prime one, by testing every possible nontrivial factor. A set which is not computable is called non-computable or non-decidable (undecidable), see for instance Post problem.

4. PRODUCTION FLOW SCHEDULING

Assuming a given SCCP framework the problem considered can be seen as an initial state and a set of priority dispatching rules generation problem which has the goal to find a multiproduct production flow timetable fulfilling production orders requirement and is feasible under production system structural constraints.

4.1 Concurrently flowing cyclic processes control

An idea standing behind of the above stated problem assumes the production flow can be realized on the set of local, e.g., shop-level executing pipeline-like flowing processes repeating same operations cyclically. The shops interact each other via the common shared resources, e.g., workers, vehicles, warehouses, and so on. That means the interacting each other shops can be modeled as elements of a SCCP. Assuming priority dispatching rules allotted to shared resources the relevant production system can be seen as controlled (i.e. synchronized) by these rules.

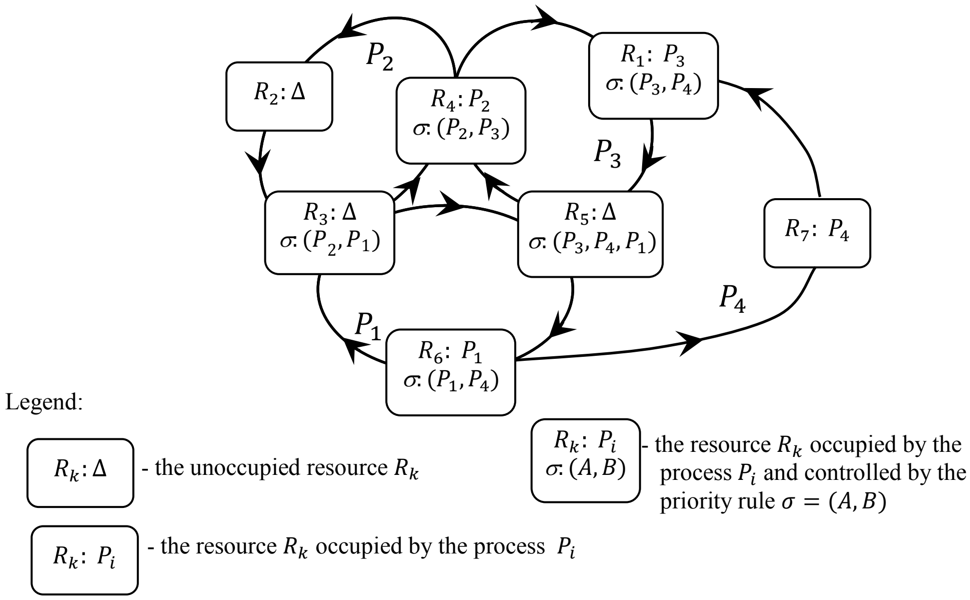


Fig. 14. Process routes structure of SCCP owning four processes

In general case different production orders can be executed along different, alternative production routes, i.e. sequences of resources (machine tools, AGVs. Warehouses, and so on). In turn, the production routes can be seen as composition of sub-parts of shop routes. So, since different rules may result in different frequency of local processes occurrence, hence different rules imply different schedules of production orders execution. Moreover, different cycles of the SCCP steady state result in different makespan of production orders. Therefore, in case of multiproduct production flow different local dispatching rules may lead to different schedules, preferring different among concurrently executed production orders.

In that context the relevant questions concerns of: What is the production system cycle guaranteeing the admissible makespan of assumed production orders? What are the dispatching rules, if any, enabling to switch the production system from a given cyclic steady state to another assumed one?

For the sake of illustration of such concurrently flowing cyclic processes control lest us consider the SCCP shown in Fig. 14. Assume an initial state and a set of priority dispatching rules as follows:

$$S^0 = (A^0, Z^0), \tag{45}$$

$$A^0 = (P_3, \Delta, \Delta, P_2, \Delta, P_1, P_4),\ Z^0 = (P_3, P_2, P_2, P_2, P_3, P_1, P_4), \tag{46}$$

and

$$\Theta = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6, \sigma_7\} \tag{47}$$

$$\sigma_1 = (P_3, P_4), \sigma_2 = (P_2), \sigma_3 = (P_2, P_1), \sigma_4 = (P_3, P_2), \sigma_5 = (P_3, P_4, P_1), \tag{48}$$
$$\sigma_6 = (P_1, P_4), \sigma_7 = (P_4).$$

From the initial state (45) the state belonging to the cyclic steady state Sc_1 the cycle of which is equal to 9 can be reached. In such steady state the state S^7 can be obtained. Changing the semaphores in this state enables to reach the new cyclic steady state Sc_1 with the period equal to 7, see Fig. 15.

That should be noted that transition between the state S^7 and S^8 does not follow the rules (12)÷(18) of the condition B. The transition considered can be seen as result of an external influence (control) aimed at changing some coordinates of the sequence Z. Such kind of states transition control is defined as follows::

Assume the states $S^k = (A^k, Z^k), S^q = (A^q, Z^q)$ are feasible: $S^k, S^q \in \mathbb{S}$. The state $\boldsymbol{S^q}$ **can be seen as a consequence of the state's** $\boldsymbol{S^k}$ **semaphore control** if the following conditions hold:

$$A^k = A^q \tag{49}$$

$$Z^k \neq Z^q \tag{50}$$

Transition between states S^k, S^q being a semaphore Z control is denoted in the following way $S^k \xrightarrow{Z} S^q$ (for instance $S^7 \xrightarrow{Z} S^8$).

For example another switch from the cyclic steady state Sc_2 to Sc_1 can be obtained by the relevant semaphores control, e.g., $S^{11} \xrightarrow{Z} S^{12}$ (Fig. 15).

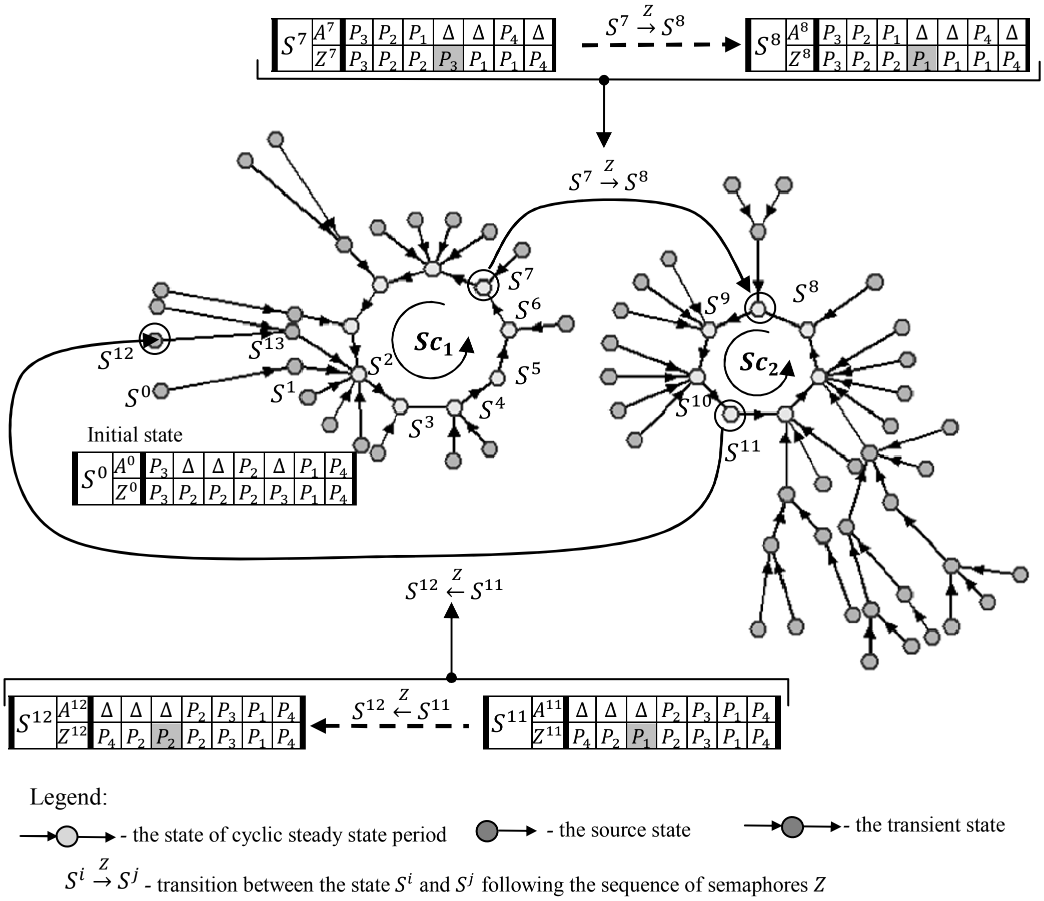


Fig. 15 An example of transition between two cyclic steady states

To summarize, transitions between cyclic steady states (e.g., Sc_1 and Sc_2) can be realized as a consequence of semaphore control as well as processes allocation and/or priority rules control. In general case such control actions should be undertaken assuming the minimal disturbances (costs) in SCCP execution. For instance, the transition $S^7 \xrightarrow{Z} S^8$ was realized in the consequence of a unique semaphore change, i.e., the semaphore associated to the resource R_4 not allotted by any process. The same processes flow distribution free transition took place in the case $S^{11} \xrightarrow{Z} S^{12}$ (see the Gantt's chart Fig. 16).

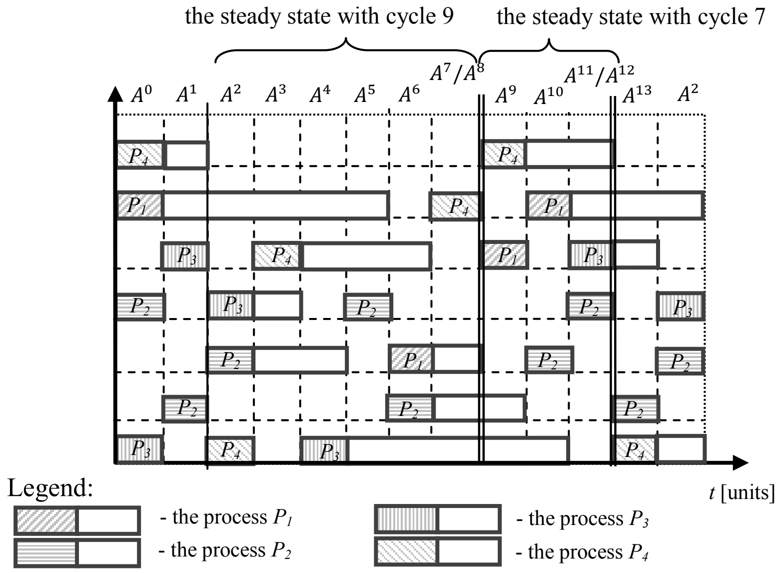


Fig. 16 Gantt's chart of the SCCP behavior corresponding with transition between two cyclic steady states

4.2 Concurrently flowing production processes control

Introduced concept of state space allows to consider different categories of states (e.g. initial and deadlock states) as well as different periods (e.g. transient and cyclic steady state). Each state determines current processes allocation at the given set of resources and semaphores determining the processes intended for direct allocation at resources in the succeeding states. It means each state specify the system resources status, i.e., determining whether the particular resource is unoccupied or occupied (either by a process waiting for access to succeeding resource or just by one executing its operation) at the moment as well as processes expected to access them the next.

In models considered the workplaces (machine tools, warehouses, etc.) or segments of AGVS transportation routes play a role of system resources. So, assuming the sequence of alternatively following each other machine tools and transportation route segments determines a manufacturing route, the manufacturing routes of two concurrently executed production flows can be modeled by the SCCP system as shown in Fig. 17.

An idea of multiproduct production flow modeling, shown in Fig. 17, assumes a given layout of manufacturing system, i.e. machine tools and AGVS as well as structure of transportation path segments (see Fig. 17a)), and the SCCP model of local manufacturing routes following sequences of alternatively occurring machine tools and paths' segments passed by AGVs (see Fig. 17b)). In case considered the manufacturing routes followed then by local cyclic processes can be seen as sequences:

$$p_1 = (sg_1, M_1, sg_2, sg_3, M_2, sg_4, sg_5, M_3, sg_6), \quad (51)$$

$$p_2 = (sg_{16}, M_4, sg_{14}, sg_3, M_2, sg_4, sg_7, sg_{10}, sg_{13}\ M_5, sg_{15}), \quad (52)$$

$$p_3 = (sg_{12}, M_6, sg_{11}, sg_{10}, sg_8, M_7, sg_9). \quad (53)$$

The manufacturing routes (51), (52), (53) describe routes of cyclic processes P_1, P_2 and P_3 of SCCP from Fig. 17.

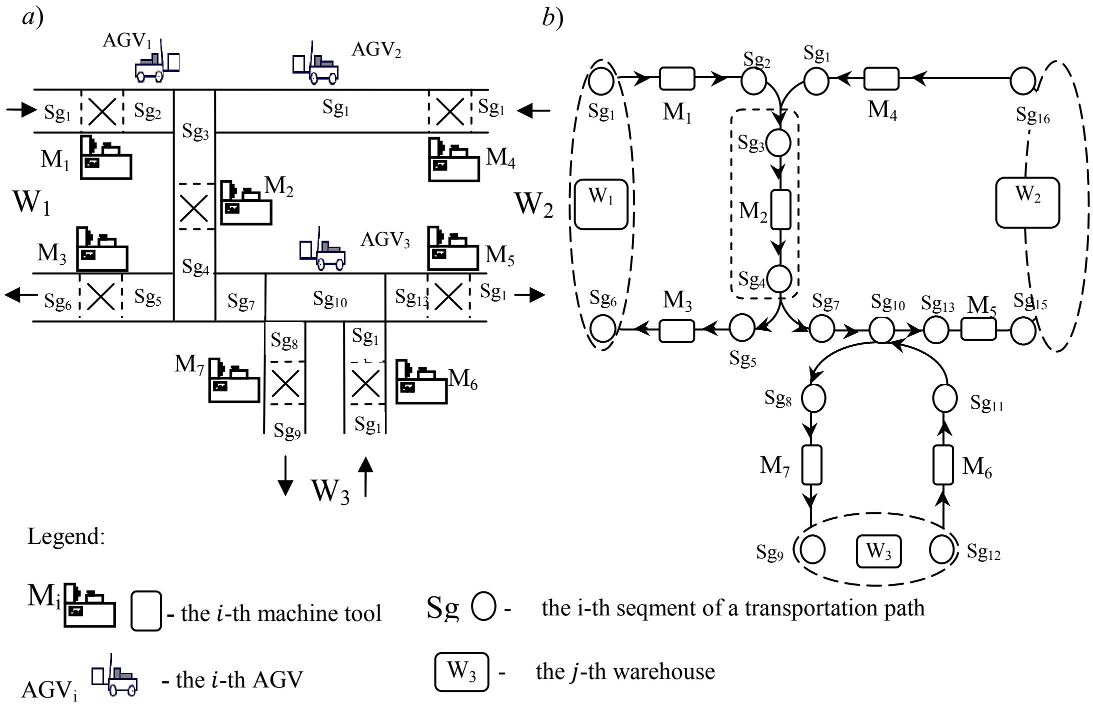


Fig. 17 Modeling the FMS in terms of SCCP. a) layout of the FMS, b)

In turn, assuming the warehouses and linked to them input and output segments of transportation routes, as well as the machine tool M_2 and associated segments sg_3, sg_4 can be treated as a kind of critical sections (distinguished by dashed lines see Fig. 17 a)), the resultant SCCP model is shown in Fig. 18.

That is easy to note that in the SCCP model from Fig. 17, besides of local cyclic processes P_1, P_2, and P_3 one can assume the new one, e.g., Pr_1 described by $pr_1 = (W, sg_1, M_1, sg_2,$ sg_3, M_2, sg_4, sg_7, sg_{10}, sg_{13}, M_5, $sg_{15}, W)$ following the model from the Fig. 17 b), i.e. the $pr_1 = (W_1, M_1$, sg_{10}, M_5, $W_2)$ following the model from the Fig. 18. This new manufacturing route is called multi-modal cyclic process and distinguished in SCCP model (see Fig. 18) by the bold dashed line.

In order to simplify further considerations let us assume the AGVs servicing particular transportation routes are disposable for any machine tool in the manufacturing routes serviced. However, they cannot deliver workpieces to the machine tool being already occupied by other workpieces. In this context, AGV_1 delivers to M_2 the workpiece processed by M_1, then after machining on the M_2 the AGV_1delivers it to M_5, and finally to W_2.

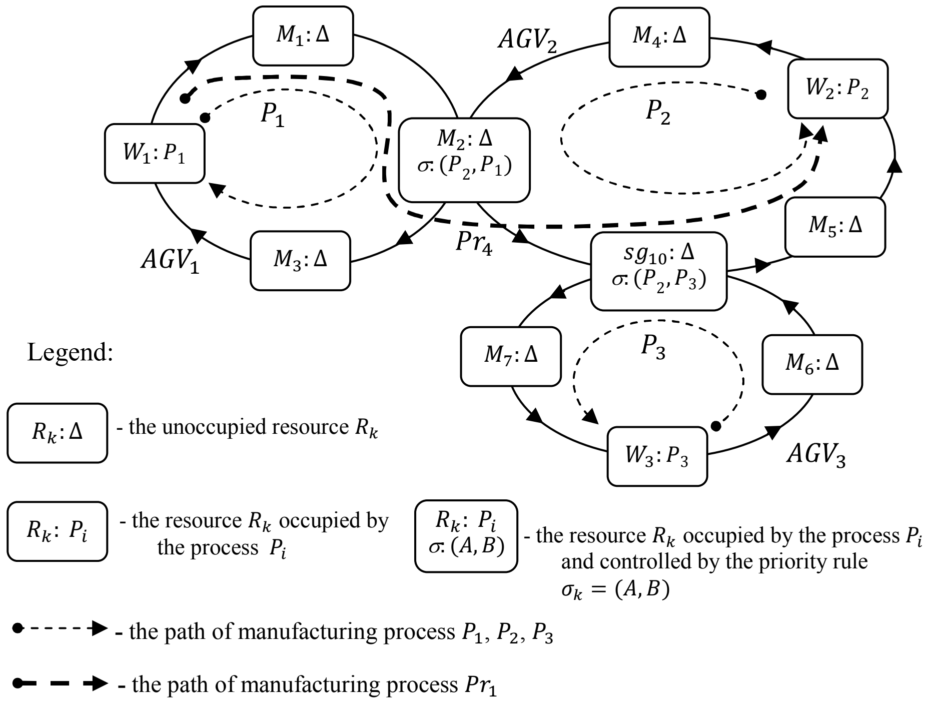


Fig. 18 The SCCP model of the FMS from Fig. 17 a)

In order to take into account such a new kind of multi-modal cyclic processes let us extend the concept of state space introduced in the Section 3.1. Consider the set of multi-modal cyclic processes each one employing resources of the more than one local cyclic manufacturing route:

$$Pr = \{Pr_1,\ Pr_2, \dots,\ Pr_{nr}\}, \tag{54}$$

where: Pr_i – the i-th multi-modal cyclic processes.

Each manufacturing processes Pr_i is specified by manufacturing route $pr_i = (R_{j_1}, R_{j_2}, \dots, R_{j_{lpr(i)}})$, $j_k \in \{1,2,\dots,m\}$, $R_{j_k} \in R$, where $lpr(i)$ denotes a length of manufacturing process route and m denotes number of resources. For example process Pr_1 from Fig. 18 is specified in the following way: $pr_1 = (W_1, M_1, M_2,\ sg_{10},\ M_5,\ W_2)$). All of the manufacturing routes belong to the set of manufacturing routes: $= \{pr_1, pr_2, \dots, pr_{nr}\}$.

That means the new, extended model of the k-th state has the form:

$$S^k = (A^k,\ Z^k, At^k), \tag{55}$$

where: A^k, Z^k – defined as in (7),
$At^k = ({at_1}^k, {at_2}^k, \dots, {at_m}^k)$ – the sequence of manufacturing process allocation, $at_i \in Pr \cup \{\Delta\}$ determines the operation from manufacturing process $Pr_i \in Pr$ executed on the resource R_i.

Consequently, the conditions (12) ÷ (18) defining $S^k \xrightarrow{B} S^q$ have to be supplemented by the following ones:

$$\forall_{i\in\{1,2,\ldots,m\}}[(a_i{}^k = \Delta) \Rightarrow (at_i{}^q = at_i{}^k)], \tag{56}$$
$$\forall_{i\in\{1,2,\ldots,m\}}[(a_i{}^k \neq \Delta) \wedge (a_i{}^q \neq \Delta) \Rightarrow (at_i{}^q = at_i{}^k)], \tag{57}$$
$$\forall_{i\in\{1,2,\ldots,m\}}[(a_i{}^k \neq \Delta) \wedge (a_i{}^q = \Delta) \wedge (at_i{}^k = \Delta) \Rightarrow (at_i{}^q = \Delta)], \tag{58}$$
$$\forall_{i\in\{1,2,\ldots,m\}}\Big[(a_i{}^k \neq \Delta) \wedge (a_i{}^q = \Delta) \wedge (at_i{}^k \neq \Delta) \wedge \big(\alpha_i(at_i{}^k) = \alpha_i(a_i{}^k)\big) \Rightarrow (at_i{}^q = \Delta) \wedge \wedge \Big(at_{\alpha_i(at_i{}^k)}{}^q = at_i{}^k\Big)\Big], \tag{59}$$
$$\forall_{i\in\{1,2,\ldots,m\}}\Big[(a_i{}^k \neq \Delta) \wedge (a_i{}^q = \Delta) \wedge (at_i{}^k \neq \Delta) \wedge \big(\alpha_i(at_i{}^k) \neq \alpha_i(a_i{}^k)\big) \Rightarrow \Rightarrow (at_i{}^q = at_i{}^k)\Big] \tag{60}$$

where: m – a number of resources,
$\alpha_i(P_j)$ – means the index of resource directly succeeding the resource R_i, in the j-th process route p_j (or manufacturing route pr_j), $\alpha_i(P_j) \in \{1,2,\ldots,m\}$.

In such extender model of SCCP state space, the new questions can be stated, e.g., Is it possible to realize production orders specified by Pr (and Πr) under constraints imposed by P (and Π) and set of priority dispatching rules Θ? What are the priority dispatching rules, if any, guaranteeing the production orders specified by Pr (and Πr) will be completed in a given time horizon *H*? It should be noted, that processes executing production orders specified by Pr (and Πr) are also cyclic ones.

Therefore, the considered problem of cyclic processes scheduling can be stated in the following way: Given a manufacturing system following modeling conditions of SCCP's state space. The multi-product scheduling problem has the goal to find a schedules which fulfill the time requirements of the production orders and is feasible under constraints imposed by the given manufacturing system.

In order to illustrate its possible instance focusing on finding up the dispatching rules enabling to consider different cyclic steady states and the sates enabling to switch from one cyclic steady sates to another lest us consider the following example.

Given SCCP model from Fig. 2. Consider initial state (20) and priority dispatching rules (21), as in Section 3.1. Assume two manufacturing processes Pr_1, Pr_2 specified by the following manufacturing routes:

$$pr_1 = (R_5, R_4, R_2, R_3), \tag{61}$$
$$pr_2 = (R_7, R_1, R_5, R_6). \tag{62}$$

In the SCCP following manufacturing routes structure from Fig. 2 the two cyclic steady states can be observed, see Fig. 4. Illustration of possible Pr_1, Pr_2 execution in the case of cyclic steady state with period $Tc = 7$ is shown in Fig. 19.

In case considered the cycle time of the process executed along the Pr_1 equals to 14 units of time, while in case of the Pr_2 equals to 7 units of time. The similar situation, however concerning the cyclic steady states with period equal to 9 is show in Fig. 20.

Note that in case of longer cyclic steady state the cycle time of the production process following the Pr_1 is shorter than in case of cyclic steady state and equals to 9 units of time. That is in cost, however of extension of the cycle time of the manufacturing process following the Pr_2, i.e. changing from 7 to 9 units of time. So, changes among different cyclic steady states may lead to significant changes of manufacturing processes periods. That means the flows of concurrently executed processes can be controlled by mean of cyclic steady state control. The work in progress and the length of the makespan can serve as objective function.

To summarize, in case the goal functions of particular simultaneously executed processes have to change the response to the following question plays a pivotal role: Does there exist a set of states allowing one to switch among given set of cyclic steady states guaranteeing the execution of production processes will free of interruptions (i.e. of delays)?

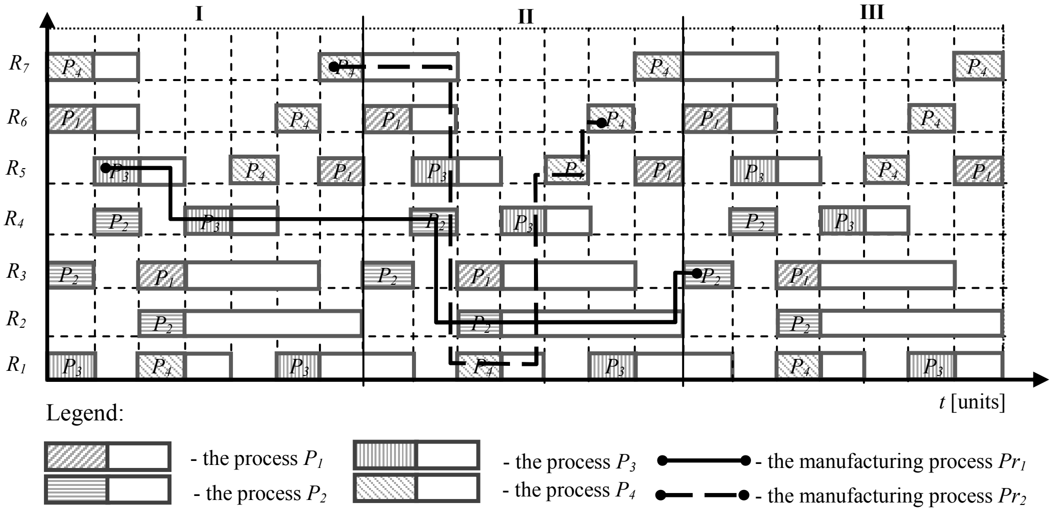


Fig. 19 Illustration of possible Pr_1, Pr_2 execution in case of cyclic steady state with the period equal to $Tc = 7$

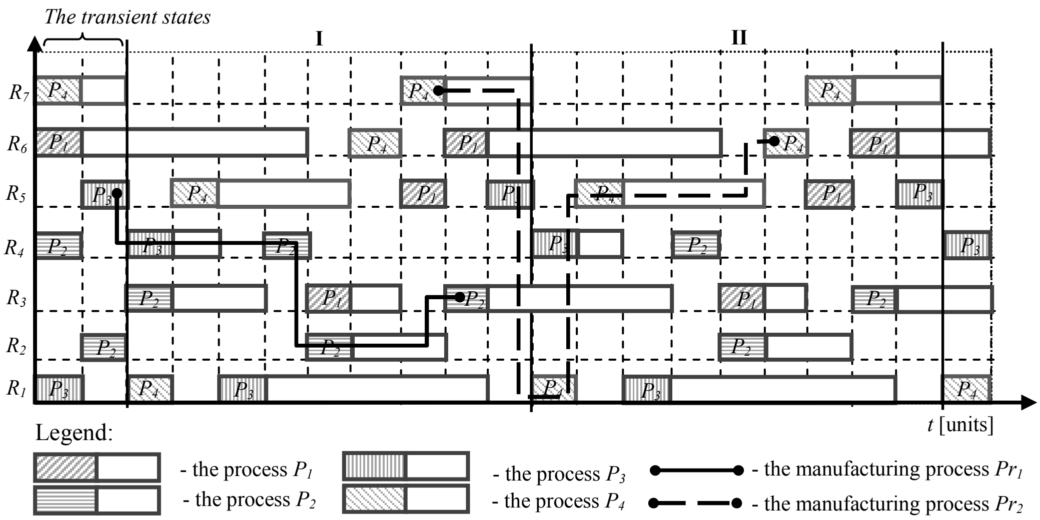


Fig. 20 Illustration of possible Pr_1, Pr_2 execution in case of cyclic steady state with the period equal to $Tc = 9$

In order to illustrate such a possibility let us consider the SCCP from Fig. 2, where for the state S^7(see Fig. 15) in the cyclic steady state of the cycle equal to 9 one is looking for switch to the cyclic steady state of the cycle equal to 7.

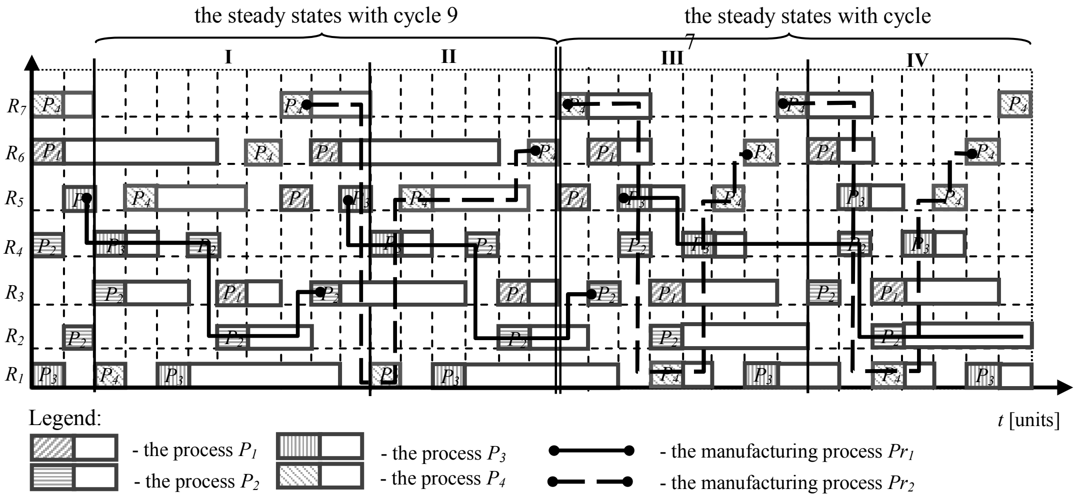


Fig. 21 Illustration of possible Pr_1, Pr_2 execution in case of switch from cyclic steady state with the period equal to $Tc = 9$ to cyclic steady state with the period equal to $Tc = 7$

Manufacturing process Pr_2 distinguished by the dashed line (see Fig. 21) completes just before its next execution in cyclic steady state length of 7. In turn, the manufacturing process Pr_1 do not change itself. The switch considered was caused by the semaphore change in the state S^7 on the Fig 15:

$$S^7 \xrightarrow{Z} S^8 \tag{63}$$

where the semaphore in the S^7 is: $Z^7 = (P_3, P_2, P_2, P_3, P_1, P_1, P_4)$, and the semaphore in the S^8 is: $Z^8 = (P_3, P_2, P_2, P_1, P_1, P_1, P_4)$.

The problem how such changes can be find, in ever, in general case still remains the open one.

From the example discussed above it follows that cyclic processes scheduling problems can be approached either in off-line or in on-line mode. In the first case, recommended by the example, the procedures supporting the decision making in the case of some routine tasks can be easily proposed. In the second case, however, any searching process has to be preceded by evaluation of possible values of the cycle period, i.e. solution of relevant set od Diophantine equations.

5. CONCLUDING REMARKS

The way an enterprise's production capacity is used decides about its competitiveness. In that context studies aimed at designing of decision support systems (DSS) dedicated to discrete processes scheduling, and especially cyclic scheduling are of primary importance. A cyclic scheduling problem is a scheduling problem in which some set of activities is to be repeated an indefinite number of times, and it is desired that the sequence be repeating.

The approach proposed based on the SCCP concept involve Diophantine equations modeling. solutions of the Diophantine equations provides evaluations of the possible cycle periods of systems considered, and then evaluation of possible makespans of concurrently executed production orders. Since Diophantine equations can be treated as a set of constraints, the constraint programming [1, 2, 3], i.e. descriptive in their character, languages can be directly implemented.

Assuming the SCCP specification encompass system structure while the set of features reflect the system behavior, one can state the following general questions:

Does there exist a control procedure enabling to guarantee an assumed system behavior subject to system's structure constraints?

Does there exist the system's structure such that an assumed system behavior can be achieved?

In this context, taking into account non decidability of Diophantine problems one can easily realize that not all behaviors can be obtained under constraints imposed by system's structure. The similar observation concerns the system's behavior that can be achieved in systems possessing specific structural characteristics. That means, the exhaustive searching for assumed control can be replaced by the step-by-step structural designing guaranteeing the required system behavior. That way, that is our believe, leads to solutions based on sufficient conditions allowing one to compose elementary systems as to obtain the final one possessing required quantitative and qualitative behavioral features.

REFERENCES

[1] ALCAIDE D., CHU CH., KATS V., LEVNER E, SIERKSMA G.: *Cyclic multiple-robot scheduling with time-window constraints using a critical path approach*. In: European Journal of Operational Research, Vol. 177, 2007, pp. 147–162.

[2] BACH I., BOCEWICZ G., BANASZAK Z.: *Constraint programming approach to time-window and multiresource-constrained projects portfolio prototyping*. In: Industrial, Engineering and Other Applications of Applied Intelligent Systems, Lecture Notes in Artificial Intelligence LNAI, Vol. 5027, Springer-Verlag, Berlin-Heidelberg, 2008, pp. 767-776.

[3] BIRGER R., EL-HOUSSAINE A., WOUT D.: *Cyclic scheduling of multiple tours with multiple frequencies for a single vehicle*. In: International Journal of Logistics Systems and Management 2009, Vol. 5, No.3/4, 2009, pp. 214 – 227.

[4] BOCEWICZ G., WÓJCIK R., BANASZAK Z., *On Undecidability of Cyclic Scheduling Problems*, In: Mapping Relational Databases to the Semantic Web with Original Meaning, Lecture Notes in Computer Science LNAI, Vol. 5914, Springer-Verlag, Berlin-Heidelberg 2009, pp. 310-321.

[5] BOCEWICZ G., WÓJCIK R., BANASZAK Z., *AGVs distributed control subject to imprecise operation times*. In: Agent and Multi-Agent Systems: Technologies and Applications, Lecture Notes in Artificial Intelligence LNAI, Vol. 4953, Springer-Verlag, Berlin-Heidelberg 2008, pp. 421-430.

[6] BOCEWICZ G., BANASZAK Z., WÓJCIK R.: *Design of admissible schedules for AGV systems with constraints: a logic-algebraic approach*. In: Agent and Multi-Agent Systems: Technologies and Applications, Nguyen N.T., Grzech A., et al. (Eds.), Lecture Notes in Artificial Intelligence 4496, Springer-Verlag, Berlin-Heidelberg 2008, pp. 578-587.
[7] CAI X., LI K.N.: *A genetic algorithm for scheduling staff of mixed skills under multi-criteria*. In: European Journal of Operational research, 125, 2000, pp. 359-369.
[8] CAMUS, H., OHL, H., KORBAA, O., GENTINA, J.-C.: *Cyclic Schedules in Flexible Manufacturing Systems with Flexibilities in operating sequences*. In: Proceedings of the 17th International Conference on Application and Theory of Petri Nets (ICATPN), 1996, pp. 97-116.
[9] CHE A., CHU CH.: *Multi-degree cyclic scheduling of a no-wait robotic cell with multiple robots*. In: European Journal of Operational Research, Vol. 199, 2009, pp. 77–88.
[10] DAVIS M., (1973). *Hilbert's Tenth Problem is Unsolvable*. In: Amer. Math. Monthly 80,233-269.
[11] ERNST A.T., JIANG H., KRISHNAMOORTHY M., OWENS B., SIER D.:. *An annotated bibliography of personnel scheduling and rostering*. In: Annals of Operations Research, 127, 2009, pp. 21-144.
[12] FOURNIER, O., LOPEZ, P., LAN SUN LUK, J.-D.: *Cyclic scheduling following the social behavior of ant colonies*. In. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2002, pp. 450-454.
[13] GAUJAL B., JAFARI M., BAYKAL-GURSOY M., ALPAN G.: *Allocation sequences of two processes sharing a resource*. IEEE. In: Trans. on Robotics and Automation, Vol.11, No. 5, 1995, pp. 748-353.
[14] GULTEKIN H., SELIM AKTURK M., KARASAN O. E.: *Cyclic scheduling of a 2-machine robotic cell with tooling constraints*. In: European Journal of Operational Research, Vol. 174, 2006, pp. 777–796.
[15] GUY, R. K.: *Diophantine Equations*. *Ch. D in Unsolved Problems in Number Theory*, 2nd ed. In: Springer-Verlag, New York, 1994, pp. 139-198.
[16] HANEN C.: *Theory and Methodology Study of a NP-hard cyclic scheduling problem: The recurrent job-shop*. In: European Journal of Operational Research, Vol. 72, 1994, pp. 82-101.
[17] HANEN, C., KORDON M.: A. *Periodic schedules for linear precedence constraints.* In.: Discrete Applied Mathematics, Vol. 157 (2), 2009, pp. 280-291.
[18] HSU T., KORBAA O., DUPAS R., GONCALVES G.: *Cyclic scheduling for F.M.S.: Modelling and evolutionary solving approach*. In: European Journal of Operational Research, Vol. 191, 2008, pp. 464–484.
[19] KATS V., LEI L., LEVNER E.: *Minimizing the cycle time of multiple-product processing networks with a fixed operation sequence, setups, and time-window constraints*. In: European Journal of Operational Research, Vol. 187, 2008, pp. 1196–1211.
[20] KORBAA, O., CAMUS, H., GENTINA, J.-C.: *A new cyclic scheduling algorithm for flexible manufacturing systems.* In: International Journal of Flexible Manufacturing Systems, Vol. 14 (2), 2002, pp. 177-191.
[21] LIEBCHEN, C., MÖHRING, R.H.: *A case study in periodic timetabling.* In: Electronic Notes in Theoretical Computer Science, Vol. 66 (6), 2002, pp. 21-34.
[22] LEVNER E., KATS V., ALCAIDE D., PABLO L:, CHENG T.C.E.: *Complexity of cyclic scheduling problems: A state-of-the-art survey*,. In: Computers & Industrial Engineering , (In print)
[23] MCCORMICK S. T., RAO U. S.: *Some Complexity Results in Cyclic Scheduling*. In: Mathl. Comput. Modelling, Vol. 20, No. 2, pp. 107-122.
[24] ODIJK, M.A.: *A constraint generation algorithm for the construction of periodic railway timetables*. In: Transportation Research Part B: Methodological, Vol. 30 (6), 1996, pp. 455-464.
[25] PINEDO M. L.: Planning and scheduling in manufacturing and services. In: Springer-Verlag, New York, 2005.
[26] POLAK M., MAJDZIK P., BANASZAK Z., WÓJCIK R.: *The performance evaluation tool for automated prototyping of concurrent cyclic processes. Fundamenta Informatice*. In: ISO Press, Editor-in-Chief, A. Skowron, Vol. 60, No.1-4, 2004, pp.269-289.

[27] SMART NIGIEL P.: *The Algorithmic Resolution of Diophantine Equations*. In: London Mathematical Society Student Text, 41. Cambridge University Press, Cambridge, 1998
[28] SPRINDZUK, VLADIMIR G.: *Classical Diophantine equations*. In: Lecture Notes in Mathematics, Vol. 1559. Springer-Verlag, Berlin 1993, 228 pp.
[29] SONG J.-S., LEE T.-E.: *Petri net modeling and scheduling for cyclic job shops with blocking*. In: Computers & Industrial Engineering,Vol.34, No.2, 1998, pp. 281–295
[30] SOON-KI HEO, KYU-HWANG LEE, HO-KYUNG LEE, IN-BEUM LEE, AND JIN HYUN PARK: *A New Algorithm for Cyclic Scheduling and Design of Multipurpose Batch Plants. In: Ind. Eng. Chem. Res., 42* (4), 2003, pp. 836–846
[31] TROUILLET B., KORBAA O, GENTINA J.-C.: *Formal Approach for FMS Cyclic Scheduling*. In: IEEE SMC Transactions, Part C, 37, Issue 1, 2007, pp. 126-137
[32] WANG, B., YANG, H., ZHANG, Z.-H.: *Research on the train operation plan of the Beijing-Tianjin inter-city railway based on periodic train diagrams.* In: Tiedao Xuebao/Journal of the China Railway Society, Vol. 29 (2), 2007, pp. 8-13
[33] VON KAMPMEYER T.: *Cyclic scheduling problems*, Ph.D. Dissertation, Fachbereich Mathematik/ Informatik, Universität Osnabrück, 2006