

A cyclic scheduling approach to maintaining production flow robustness

Bocewicz, Grzegorz; Nielsen, Izabela; Banaszak, Zbigniew; Majdzik, Pawel

Published in:
Advances in Mechanical Engineering

DOI (link to publication from Publisher):
[10.1177/1687814017746245](https://doi.org/10.1177/1687814017746245)

Creative Commons License
CC BY 4.0

Publication date:
2018

Document Version
Publisher's PDF, also known as Version of record

[Link to publication from Aalborg University](#)

Citation for published version (APA):
Bocewicz, G., Nielsen, I., Banaszak, Z., & Majdzik, P. (2018). A cyclic scheduling approach to maintaining production flow robustness. *Advances in Mechanical Engineering*, 10(1).
<https://doi.org/10.1177/1687814017746245>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy

If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

A cyclic scheduling approach to maintaining production flow robustness

Grzegorz Bocewicz¹, Izabela Nielsen², Zbigniew Banaszak¹ and Pawel Majdzik³

Abstract

The organization of flow production, which is typical found in assembly processes, involves a repetitive, fixed-takt time flow of same-size production batches. The cyclic nature of the production flow, which ensures a steady production rhythm, enables just-in-time planning and organization of the associated supply chains. Disruptions in the operation of machinery and equipment, which occur in practice, lead to deviations from nominal operation times. These types of local disturbances lead to changes in production takt time, making it necessary to adjust previously created schedules for delivery/reception of materials and products. Assuming that a control action can be taken to adjust transport operation times within a specified time range, the problem of cyclic scheduling of production flows boils down to seeking conditions the satisfaction of which will guarantee robustness to this kind of disruptions. Satisfaction of robustness conditions allows a return to the nominal production takt time and appropriate adjustment of the production flow trajectory (which makes it possible for the system to return to the previously scheduled delivery times). Numerous examples are included to illustrate the principles of the proposed research methodology aimed at finding solutions for robust scheduling of fixed-takt time production flow.

Keywords

Cyclic steady state, reachability, production flow robustness, multimodal process, production takt time

Date received: 31 August 2017; accepted: 10 November 2017

Handling Editor: Peter Nielsen

Introduction

Cyclic scheduling is known to be one of the most effective methods of operational planning in transport and production systems.¹ In transport systems, this approach is predominantly used in areas related to the transportation of people, including rail transport, urban transport, intercity bus transport and so on. Rhythmic delivery, repeated at regular intervals, is also a feature of systems of cyclic delivery of goods, such as food products or consumables, to distribution centres. In production systems, in turn, cyclic schedules play the role of specific production flow patterns (scenarios) in steady-state situations (i.e. excluding states associated with production start-up and shutdown). In this

context, it is also worth highlighting the relationships that exist between a clock-face schedule or a cyclic schedule and a timetable, which is a list or a table of events arranged according to the time when they take

¹Faculty of Electronics and Computer Science, Koszalin University of Technology, Koszalin, Poland

²Department of Materials and Production, Aalborg University, Aalborg, Denmark

³Institute of Control and Computation Engineering, University of Zielona Góra, Zielona Góra, Poland

Corresponding author:

Grzegorz Bocewicz, Faculty of Electronics and Computer Science, Koszalin University of Technology, Sniadeckich 2, 75-453 Koszalin, Poland. Email: bocewicz@ie.tu.koszalin.pl



place, for example, train departures/arrivals or takt-based product mix flows.

Cyclic processes are also connected with the concept of multimodal processes, with the latter being defined in terms of the former.²⁻⁶ Multimodal processes, that is, processes that use other processes to accomplish their goals, are commonly found in day-to-day practice, among others in shop floor transport systems in discrete production systems, computer networks for data transmission, power transmission networks, urban transport systems, digital communication systems and so on. The best way to get an intuitive understanding of what a multimodal process is, is to refer to the example of passenger transport in a rapid transit (metro) network. Metro lines (local cyclic processes) connect at transfer stations to form a transport network that allows passengers to follow their own itineraries (routes). A passenger travelling by metro usually uses several different local lines, and his or her movement can be interpreted as a multimodal process, that is, a process that is executed using parts of various different local processes. The execution of the processes that represent the behaviour of the individual metro lines, associated with concurrently executed local processes, depends solely on the adopted synchronization mechanisms that determine access to shared workstations, such as interchanges. This means that cyclic behaviour of multimodal processes depends on the cyclic behaviour of the environment in which they are executed, that is, the so-called system of local concurrent cyclic processes (SCCPs).^{4,6,7}

The problems of reachability of cyclic steady states considered in an SCCP environment fall into the category of cyclic scheduling problems, and in particular the problems associated with planning of concurrent multimodal processes (of different nature and character). As such, they are of vital importance to the performance of systems for production, urban transport, digital communications and so on. Literature provides a wealth of algorithms for determining optimum operation schedules dedicated to different work assumptions and work environments (e.g. timetabling tasks,⁸ telecommunications transmissions,⁹ or production planning^{5,6,10-13}). Some of the most important cyclic scheduling problems are optimization problems, such as the basic cyclic scheduling problem (BCSP)^{1,14} and its extensions associated with scheduling in production job-shops (so-called job-shop problems): the general cyclic job-shop problem,⁹ the cyclic flow-shop problem^{1,8} and the cyclic open-shop problem.¹⁵ With the exception of BCSP, all the problems listed above belong to the class of NP-hard problems, which means their solution requires the use of artificial intelligence methods for any real-sized problems.^{1,16}

As mentioned above, the cyclic environment of an SCCP implies that the multimodal processes executed

in it exhibit a cyclic behaviour. By limiting the discussion to discrete manufacturing systems, it is easy to note that production flows in such systems are a model example of a multimodal process. Analogies can be found when one looks at the flow of individual products and/or product batches that are carried by different devices across different workstations (machining, washing, inspection, assembly, etc.). To say that a multimodal process (i.e. a production flow modelled by it) is a cyclic process is to say that it is a steady-state process or more precisely a cyclic steady-state process. It is worth recalling here that a production system or a production flow is in a steady state if the variables which define the behaviour of the system or the process, that is, its production takt time and/or cycle period, are unchanging in time.

The basic organizational tenet of this type of rhythmic production is the concept of cyclic schedules, which is an indefinitely repeated sequence of operations (tasks) executed on a set of workstations. In most real production settings, however, cyclic schedules are disrupted by a variety of different unexpected events. The occurrence of a disturbance may interrupt system operation and/or upset the previously established schedule. Such disruptions are generally difficult to take into account/anticipate while generating a schedule. Thus, it is necessary to develop predictive/reactive schedules which can accommodate disruptions while maintaining high shop performance.^{17,18}

The disruptions in the operation of machinery and equipment occurring in practice lead to deviations from nominal operation times. These local disturbances change production takt time, making it necessary to adjust the previously created cyclic schedules. Assuming that a control action can be taken to adjust transport operation times within a specified time range, the problem reduces to seeking conditions which, when satisfied, will guarantee robustness to disruptions understood as the ability of the system to return to the nominal cyclic state. Such conditions guarantee a return to the nominal cyclic steady state (with a specified production takt time and cycle period) and appropriate adjustment of the production flow trajectory (allowing a return to scheduled delivery times).

Given the definition of robust predictive/reactive scheduling as being focused on generating predictive/reactive schedules oriented towards minimizing the effects of disruption on schedule performance, our goal in this study was to investigate robust predictive/reactive scheduling with regard to job-shop scheduling problems connected with random changes in operations executed on automated guided vehicles (AGVs) and/or workstations. Accordingly, in this article, a robust reactive scheduling method was developed for solving job-shop cyclic scheduling problems in a dynamic environment. This method uses an algorithm

for re-scheduling the affected operations and uses a procedure owing to which the additional idle time is incorporated into the schedule to accommodate the impact of unexpected disruptions. The performance of the proposed scheduling method is then compared with a method based on the approach presented by Seybold et al.¹⁹

The remainder of this article is organized as follows: section ‘Related work’ provides a brief overview of related works. Section ‘Cyclic steady states’ reachability’ presents the fundamentals of cyclic steady states’ reachability modelling. Section ‘Problem statement’ gives the problem statement and presents a dedicated declarative model. Section ‘Robust production flow schedule planning’ addresses the main concept of a cyclic scheduling approach to maintaining production flow robustness and provides examples illustrating the proposed scheduling method. Finally, section ‘Concluding remarks’ offers some concluding remarks.

Related work

Numerous papers address the problems of cyclic (periodic) scheduling in flexible manufacturing. This approach is recognized as an effective way to process various manufacturing, computing and transportation processes, including those where set-up and transportation times are relevant^{7,12,15,20,21}). The well-recognized advantages of cyclic scheduling over static (non-cyclic) scheduling include, in particular, more efficient material handling, better station utilization and simpler shop floor control. Some other examples of multi-AGV cyclic scheduling have been studied as part of modelling frameworks regarding the periodic vehicle routing problem (VRP), in which delivery routes are constructed over a period of time,^{22,23} including instances of VRP taking into account constraints imposed by time windows²⁴ and models of multimodal processes flow²⁵ as well as multi-cyclic flow shop scheduling, which is basically an extension of this last type of models. Multi-cyclic scheduling is a generalization of common cycle scheduling in which the cycle time for each product is required to be an integer multiple of the baseline cycle time.²⁰ It should be noted that a majority of research in the field focuses either on scheduling available transport modes so that they can service customers in specific time windows or on designing supply networks while taking into account the size and capacity of the planned fleet as well as the topology and traffic capacity of routes. Relatively few studies are devoted to frameworks that combine path routing and AGV fleet scheduling while providing an integrated and unified approach to the modelling and design of cyclic production flow schedules that would guarantee efficient material handling, better workstation utilization and simpler shop floor control.¹² By reason of the

NP-hard character of considered problems, well-known approaches following the source range from dynamic optimization methods²⁶ through declarative models (based on constraint satisfaction problems – (CSPs))^{27,28} to heuristic-driven genetic algorithms.^{23,29}

A separate class of cyclic scheduling problems encompasses instances of production systems in which data (and, in general, decision variables) are only known approximately with low accuracy (i.e. their value is uncertain or fuzzy). The topic of processes flow planning subject to fuzzy windows time constraints modelled as a fuzzy CSP is discussed by Bocewicz et al.^{5,6}), where the approach of the step-by-step composition/construction of a multimodal transportation network built from so-called elementary sub-structures is proposed. The sub-structures are material handling systems composed of vehicles which move materials between machines placed along different modes of cyclic guided routes in a model of system composed of trains/risers/elevators. The approaches to this class of problems proposed in literature include the previously mentioned meta-heuristic-driven evolutionary and constraint programming methods. It is worth noting that a holistic approach allowing interactive prototyping of alternative solutions³⁰ involves the implementation of a declarative-modelling-driven method. This approach that can be seen as continuation of our former works^{6,31} allows one to treat the above-mentioned scheduling problem in terms of frequently asked questions: What schedule of an AGV fleet following a given set of operation times maximizes operation availability and productivity? or a reverse way – What set of operation times can guarantee that the resultant schedule of a given AGV fleet maximizes both operation availability and productivity?

Since ‘optimal’ schedules, once deployed, are affected by irregularities and are far from optimal in practice, one has to build robustness into the schedule during the schedule design process.^{18,32} In this context, a relatively new aspect of cyclic scheduling research are robust scheduling problems.^{18,33,34} The aim of this new approach is to develop robust solutions capable of accommodating disruptions during schedule execution. In this study, robustness is understood as the ability of a system to resist change without the need to adjust its initial stable configuration. In this context, a proactive (robust) schedule is designed to protect a baseline schedule developed prior to the start of production flow from disruptions. Reactive scheduling, on the other hand, refers to procedures that may be used to revise or re-optimize the baseline schedule when unexpected events occur. The focus of this article is a deadlock-free proactive/reactive cyclic scheduling approach, which can be used to react to the occurrence of a variety of disruptions in flexible job-shops in real time. In general, internal and external disruptions such

as machine breakdown, process time variation, an urgent job, arrival of a new job order, or order cancellation are of main interest.

To study robust predictive/reactive scheduling in a job-shop with random disruptions in AGV and/or workstation operation time, one first has to choose a modelling framework for developing robust scheduling methods. As it is easily seen, the class of discrete production systems considered in this article includes structures comprising transportation robots, workstations and input and output buffers. Such components are typical of manufacturing processes belonging to the class of discrete-event systems (DESS).⁷ Of course, since DESSs are discrete-state, event-driven systems, their state evolution depends entirely on the occurrence of discrete events over time. There are many different modelling techniques for DESSs, such as Petri nets, extended state machines, event-graphs, formal languages, generalized semi-Markov processes, non-linear programming, automata and computer simulation models (see, for example, studies^{35–37} and the references therein). An emerging approach in this area is offered by the max-plus algebra framework. Max-plus algebra can be used to model and analyse a production system within the linear framework,³⁸ allowing, for example, the use of Petri nets to simulate system behaviour. The major advantage of this framework is that it eliminates the need to formulate scheduling problems as non-linear optimization problems, which is the main drawback of the classical algebra framework applied to such tasks.¹⁹

Cyclic steady states' reachability

Figure 1(a) shows an example of a battery assembly system¹⁹ consisting of five workstations and two industrial trucks (W_1 and W_2) which travel cyclically along closed-loop transport routes. Production flow routes and shop floor transport routes (between workstations) are marked with red and blue lines, respectively.

From the system in Figure 1(a), it is easy to see that the material flow in this system can be modelled by the solution in Figure 1(b), with the long routes travelled by industrial trucks (across several workstations) in Figure 1(a) (W_1 , W_2) being replaced by local route sequences of Figure 1(b) (where the directions of local routes of the industrial trucks: W_1 , W_2 , W_3 , W_4 , W_5 , W_6 are the same as the directions of the routes shown in Figure 1(a)). It is assumed that the corresponding times of transport between workstations in both versions of the system are the same and that the reversing times of the industrial trucks are negligibly short. An approach like this emphasizes the possibility of using various local means of transport (e.g. AGVs, robots, conveyors) and also takes into account different times of transport between workstations.

The flow production system considered here, in which shop floor transport is based on local transport lines (allowing for the transportation of products along cyclic routes), can be modelled as a system of concurrent cyclic processes (see Figure 2). In this system, work units move along fixed routes of so-called multimodal processes (red line $-mP'_1$ and mP''_1), that is, processes responsible for production flow, whose routes are a combination of segments of local transport process routes. In the case considered here (Figure 1(b)), work units are transported along fixed routes by six self-propelled industrial trucks, whose operations are modelled as local processes (P_1 , P_2 , P_3 , P_4 , P_5 and P_6).

In general, it is assumed that the class of SCCP includes two types of processes:

- *Local processes* (representing modes of transport $-P_1$, P_2 , P_3 , P_4 , P_5 , P_6), whose operations are cyclically repeated along fixed routes (sequentially used workstations). For the system of Figure 2, the route of process P_1 (process representing operations of truck W_1) has the following form: $p_1 = (R_6, R_8, R_1, R_9)$. Local-process routes are defined in the following way

$$p_2 = (R_1, R_{12}, R_2, R_{13}),$$

$$p_3 = (R_2, R_{17}, R_4, R_{16}), \quad p_4 = (R_7, R_{10}, R_3, R_{11})$$

$$p_5 = (R_3, R_{14}, R_4, R_{15}), \quad p_6 = (R_4, R_{18}, R_{19}, R_4)$$

where the i th operation (executed on workstation R_k) of process P_j is designated as $o_{i,j}$, and $t_{i,j}$ stands for the time over which this operation is executed.

- *Multimodal processes* (representing production flow mP_1), whose operations are cyclically carried out along routes that are combinations of segments of local-process routes. For the system in Figure 2, the route of multimodal process mP_1 consists of two segments

$$mp_1 = (mp'_1, mp''_1)$$

$$mp'_1 = ((R_6, R_8, R_1), (R_1, R_{12}, R_2), (R_2, R_{17}, R_4))$$

$$mp''_1 = ((R_7, R_{10}, R_3), (R_3, R_{14}, R_4), (R_4, R_{18}, R_{19}))$$

where the i th operation (executed on workstation R_k) of multimodal process mP_j is designated as $mo_{i,j}$, and $mt_{i,j}$ and $md_{i,j}$ stand for the times over which it is executed (where a distinction is made between $mt_{i,j}$, which refers to transport operations, and $md_{i,j}$, which refers to station operations).

Operations in processes of this sort are executed on two types of workstations: *process-specific workstations* (each of which is used by only one process of a given sort $-R_5 - R_{19}$) and *shared workstations* (each of which is used by more than one process of a given

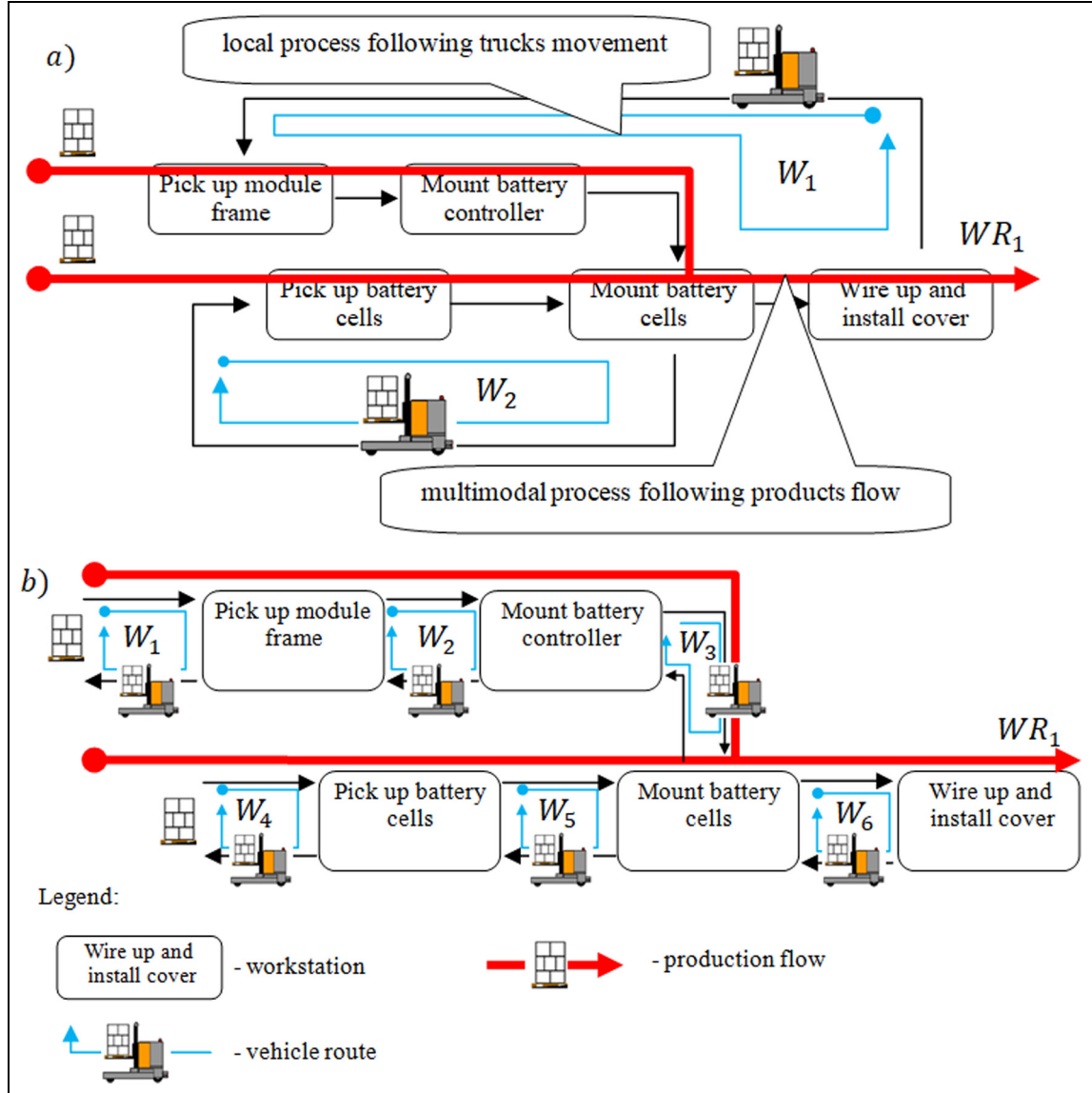


Figure 1. Example of an assembly process¹⁹ with (a) a transport subsystem consisting of two industrial trucks and (b) a transport subsystem consisting of six industrial trucks.

sort $-R_1 - R_4$). Workstations represent the individual elements of the structure of a production system. In the case considered in this study, $R_1 - R_5$ represent workstations, $R_6 - R_7$ represent input buffers and $R_8 - R_{19}$ correspond to transport sections/paths travelled by self-propelled trucks.

Local processes use shared workstations in a mutual exclusion mode, that is, only one process can be performed on a workstation at a given time. Shared local-process workstations are accessed in the order specified by priority dispatching rules Θ . It is assumed that $\Theta = \{\sigma_1, \dots, \sigma_k, \dots, \sigma_{lk}\}$, where σ_k is a sequence whose elements specify the order in which local processes receive access to workstation R_k . For example, $\sigma_4 = (P_3, P_5, P_6)$ means that the order of access to

workstation R_4 is process P_3 followed by P_5 followed by P_6 , and over again.

The next operation of a process starts immediately after the current operation has been completed provided the workstation required to execute it is available (it is not occupied and the appropriate priority dispatching rules allow access to the workstation). While waiting for an occupied workstation, the process does not release the workstation assigned to the previous operation.⁴ Moreover, processes are assumed to be non-preemptive, and the order in which operations are performed does not depend on external disruptions. External disruptions may, however, affect workstation operation times (operations of multimodal processes on workstations $R_1 - R_5$) marked in Figure 2 as $md_{i,j}$.

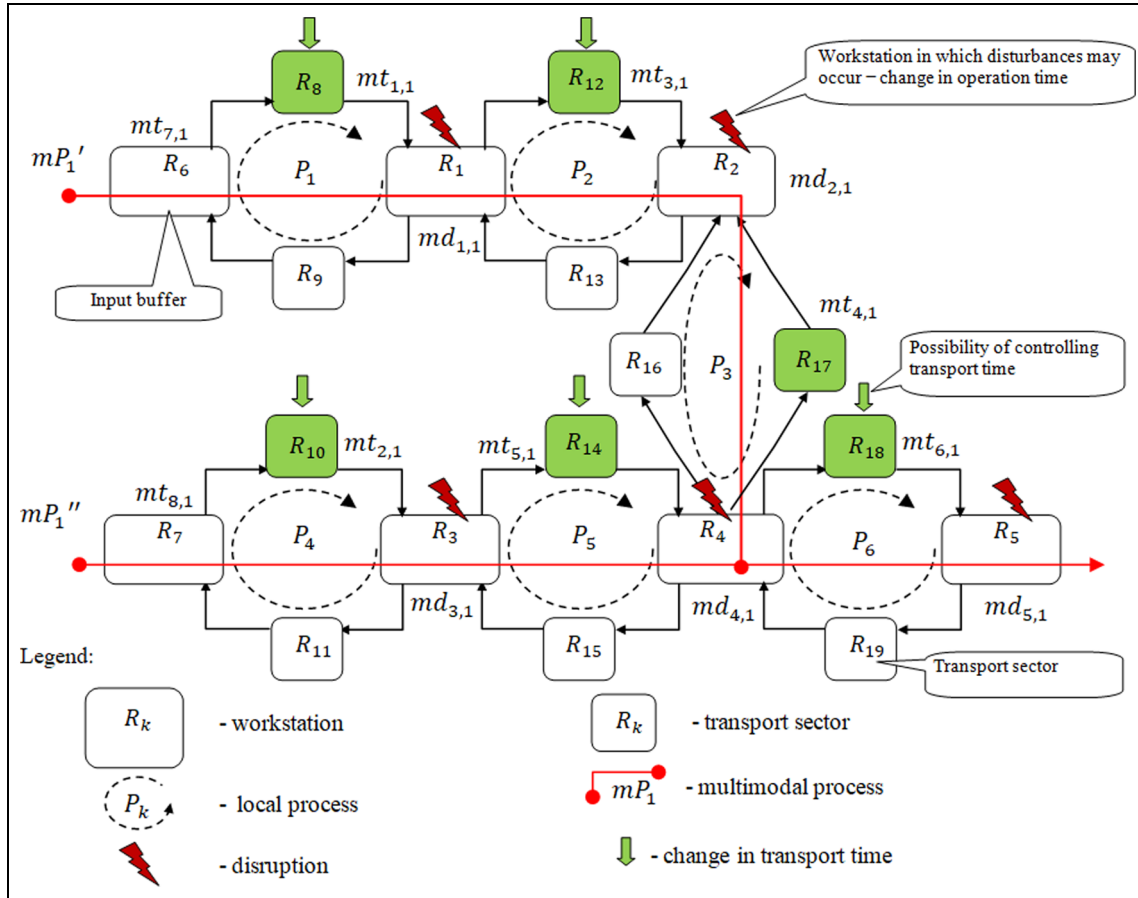


Figure 2. An SCCP for the system of Figure 1(b).

Source: Adapted from Seybold et al.¹⁹

In this approach *disruption* Z of a system is understood as a change in the value of one or more workstation operation times which affects the course of other operations executed in that system.

The above parameters describing the type and number of system workstations R (assembly stations, path sectors) used the local P /multimodal mP processes executed in the system (along the given routes p/mp) and the operations $o_{i,j}/mo_{i,j}$ occurring in those processes, as well as a set of priority dispatching rules Θ and the values of operation times make up the *structure* of an SCCP. In other words, whenever the expression 'structure of an SCCP' is used in the text, depending on the context, it will either emphasize the topological features of the structure of an SCCP (e.g. transport and/or production routes) or a combination of values of the parameters describing the elements of this structure. Similarly, the *behaviour* of an SCCP system will be understood as a set of its possible responses (sequences of states and events) activated by potential initial-state values (initial allocations of processes on workstations). The various behaviours of an SCCP, corresponding to the respective sequences of states and events (related to

the execution of system operations), are represented by cyclic schedules defining the successive.

The different operation times and dispatching rules for synchronization of access of local transport processes to shared production workstations generate different cyclic states of local and multimodal processes, that is, different cyclic behaviour of an SCCP. Figure 3 shows examples of cyclic schedules for the execution of local processes in a material handling network (Figure 3(a)) and multimodal processes in a production flow network (Figure 3(b)) for the SCCP in Figure 2. The nominal operation times are shown in Table 1.

The columns of the cyclic schedules shown in Figure 3 correspond to the states occurring in the closed-loop trajectories of cyclic steady states. In the case illustrated in Figure 3, it is assumed that events (characterizing states) corresponding to the individual columns of the schedule occur in cycle time; in other words, it is assumed that all events take place in cycle time expressed as symbolic units of time (t.u.). In general, the number of occurrence of events differs depending on the operations associated with those events and are multiples of symbolic time units. This means, in

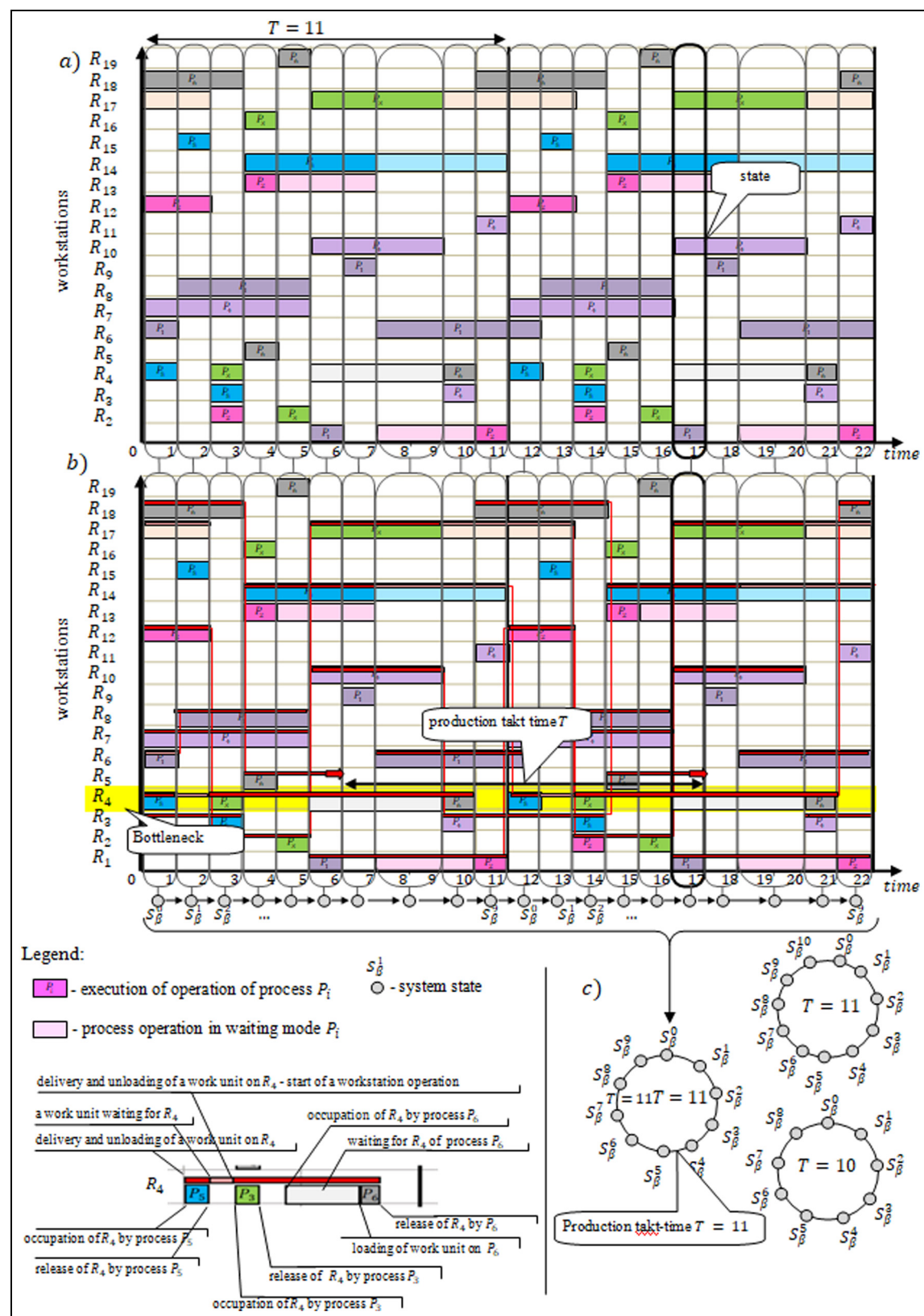


Figure 3. Operation execution schedules for SCCP from Figure 2 for (a) local processes, (b) multimodal processes and (c) the space of cyclic steady states reachable in the SCCP.

Table 1. Admissible changes in operation times of processes executed in the SCCP of Figure 2.

	Nominal time	Time interval		Nominal time	Time interval
$md_{1,i}$	6	[4,7]	$mt_{1,i}$	4	[3,5]
$md_{2,i}$	3	[3,4]	$mt_{2,i}$	4	[3,5]
$md_{3,i}$	5	[4,6]	$mt_{3,i}$	2	[2,3]
$md_{4,i}$	8	[6,10]	$mt_{4,i}$	4	[3,5]
$md_{5,i}$	3	[3,4]	$mt_{5,i}$	4	[3,5]
			$mt_{6,i}$	4	[3,5]
			$mt_{7,i}$	5	[5,5]
			$mt_{8,i}$	5	[5,5]

particular, that a scheduling period calculated in symbolic units of time does not determine the number of states occurring in the cyclic steady state that models it (cf. Figure 3(a) and (c)).

It should be noted that the typical parameters characterizing the behaviour of an SCCP, that is, the system's period α and production takt time T , are governed by the bottlenecks that may occur on overloaded workstations or local processes (when transport time is longer than workstations' operation time). In the considered case, the bottleneck is workstation R_4 . This workstation, which is dedicated to the 'mount battery' operation, is only made available in periods when work units are moved between operations (loading and/or unloading).

The operation execution times for processes of the SCCP in Figure 2 can change due to a variety of disruptions Z that can occur in flexible job-shops in real time. In the general case, internal and external disruptions such as workstation and/or AGV failures, process time variation, an urgent job, arrival of a new production order and order cancellation are considered.

In the case analysed here, however, only internal disruptions regarding workstation operation times $md_1 - md_5$ and/or external disruptions related to adjustment of transport operation times $mt_1 - mt_6$ are taken into account. Consequently, it is assumed that process disruptions are reflected in the elements of a cyclic steady states' space (and consequently in the corresponding cyclic schedules of admissible production flows). An example of a space of admissible cyclic steady states is shown in Figure 4. In general, in a given SCCP, multiple cycles comprising its steady-state space (i.e. a space of admissible steady-state behaviours) can be reachable. Families of admissible cyclic steady-state spaces (the individual layers shown in Figure 4) are generated for given instances of SCCP structures $-SC_i$ (for $i = \beta, \dots, \omega$) – which differ from each other in the values of the variables $md_{i,j}$ and $mt_{i,j}$ (specified in Table 1 – it is assumed that the priority dispatching rules are identical).

The successive layers thus correspond to the successive changes in the structure of the SCCP. In summary, for each set of potential disruptions, that is, a set of

possible changes in the structure of an SCCP, and, consequently, for each SCCP model associated with it, there exists an achievable behaviour that is part of a family of cyclic steady states. The individual cyclic states comprising a given family correspond to instances of ordered triples: a set of priority dispatching rules, initial state and operation execution times. The initial state is characterized by a vector of pairs governing the process currently allocated to a workstation and the next process allocated to the same workstation.

In the adopted model, each disruption Z in production flow knocks the system out of the currently executed steady state, leading to a transition to a cyclic steady state which is a component of a different layer. Transition between the successive steady states is made possible by a transient state executed in a direct or an indirect mode.

A direct transient state between two steady states of the same layer of a cyclic steady-state space exists for states in which the same processes are currently allocated to the same workstations, and the subsequent processes waiting for allocation to these workstations are different. A transition between two consecutive steady states is an effect of a changeover to a different process allocated to the same workstation (in conformity with a given dispatching rule). In turn, an indirect transient state comprises a chain of states in which the first one belongs to the given steady state, whereas all the subsequent states, reached as a result of a changeover to a different process, do not. The last state of the chain belongs to a cyclic steady state reachable from the initial state (beginning the changeover).

Direct/indirect transient states which form bridges between cyclic steady states belonging to different layers are executed in an analogous way, but they arise as a consequence of changing the times of local/multi-modal process operations. A direct transient state between two cyclic steady states belonging to different layers of a cyclic steady states' space exists for states in which the same processes are currently allocated to the same workstations and the subsequent processes waiting for allocation to these workstations are also identical. The transition between two consecutive steady states is a consequence of changing the operation times

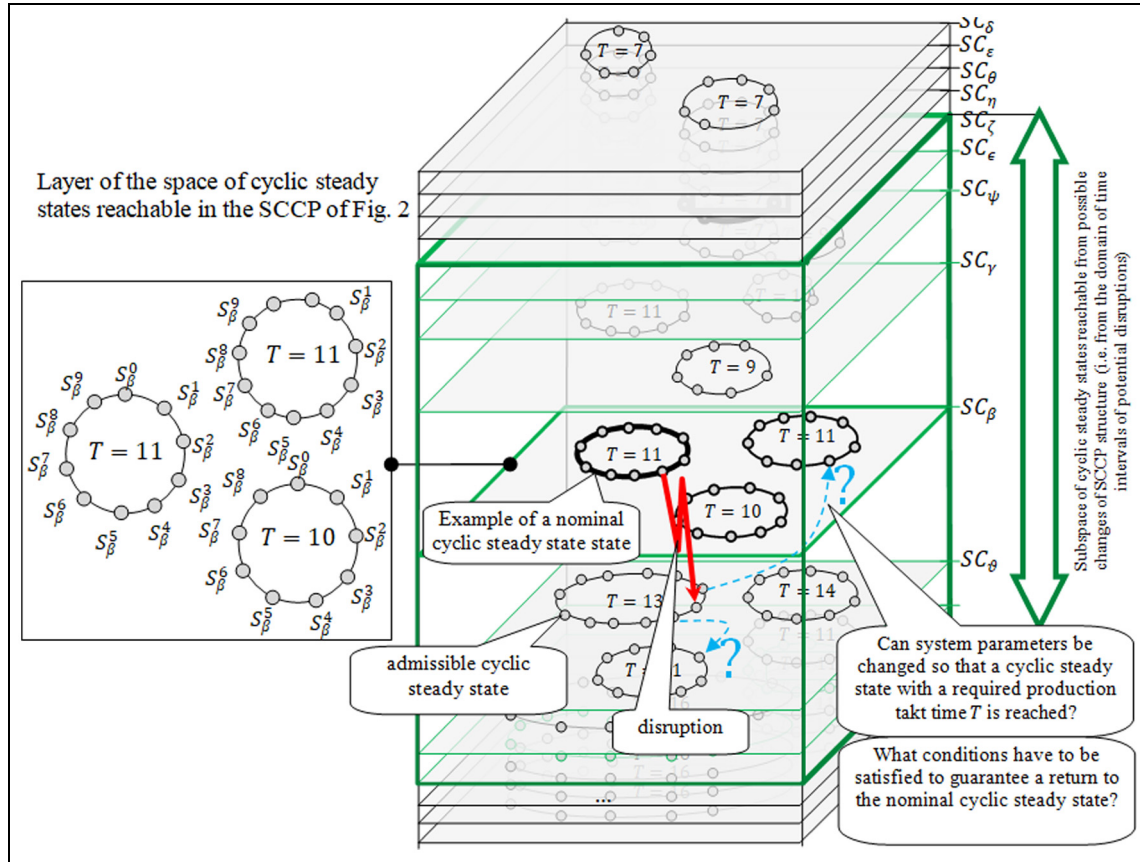


Figure 4. Space of admissible cyclic states of the SCCP of Figure 3.

of the relevant processes. On the other hand, an indirect transient steady state is composed of a chain of states in which the first one belongs to the given steady state, whereas all the subsequent states, reached as a result of changing the operation times, do not. The last state of the chain belongs to a steady state reachable from the initial state.

A vital question that arises in this context is the decidability of the reachability problem, that is, whether, for a given cyclic steady state, there exists a transient state leading to another, arbitrarily selected cyclic steady state of the same space. To answer this question, it is necessary to search for conditions that when satisfied guarantees the existence of a transient state connecting the selected cyclic steady states (generally belonging to different layers). In other words, the conditions sought are the conditions for the reachability of a specific cyclic steady state from an arbitrarily selected previously given state.

Problem statement

The problems regarding SCCP discussed in the literature of the subject usually fall into two groups: problems of *analysis* of behaviour attainable in a given

system and problems of *synthesis* of a system structure guaranteeing the occurrence of a given behavior.⁴ The former group of problems involves answering the following question:

- Whether it is possible in a given structure of an SCCP to reach cyclic steady states that would meet the expectations regarding timely execution of processes (order completion date, production takt time, production period, etc.)?

An instance of the latter class of problems is the following question:

- Does there exist a system structure (defined by the values of parameters such as the times of workstation and transport operations) that would guarantee the reachability of given cyclic states (characterized by timely execution of processes, a specific production takt time and period, etc.)?

The case considered in the previous section can be formulated as the following problem:

Given is an SCCP with a structure as shown in Figure 2, with workstation operation times (md_i) and transport operation times (mt_i) as shown in Table 1. Moreover, known are nominal cyclic states (reachable for the nominal times given in Table 1, i.e. in a situation when there are no disruptions) and the corresponding execution schedules (Figure 3). Given are also the quantitative characteristics of these states described by production takt time T and production period α . Assuming that a cyclic steady state reached as a result of permanent disruption Z causing a change in the execution time of one or more workstation operations ($md_{1,1} - md_{5,1}$, where admissible changes fall within the ranges given in Table 1) is known, the goal is to find an answer to the following question:

- Can transport operation times $mt_{1,1} - mt_{8,1}$ be changed so that the system can again reach (return to) the previous cyclic steady state

characterized by a given production takt time T^* and production period α^* ?

The problem formulated in this way is a specific case of the *synthesis* problem, the goal of which is to seek an SCCP structure that can ensure the reachability of a given cyclic steady state. It should be stressed that the existence of a structure and the associated cyclic state that can accommodate the effects of the disruption is a necessary condition for the system under consideration to be able to return to the nominal production takt time T and period α . An example of a disruption that knocks a system out of its nominal cyclic steady state is shown in Figure 5 (Figure 5(a)– SCCP model, Figure 5(b)– schedules before and after disruption, Figure 5(c)– cyclic steady states).

This is a case of a direct transition between cyclic states of one family (layer) of an admissible-states' space. This transition is made possible by states S_β^5 and

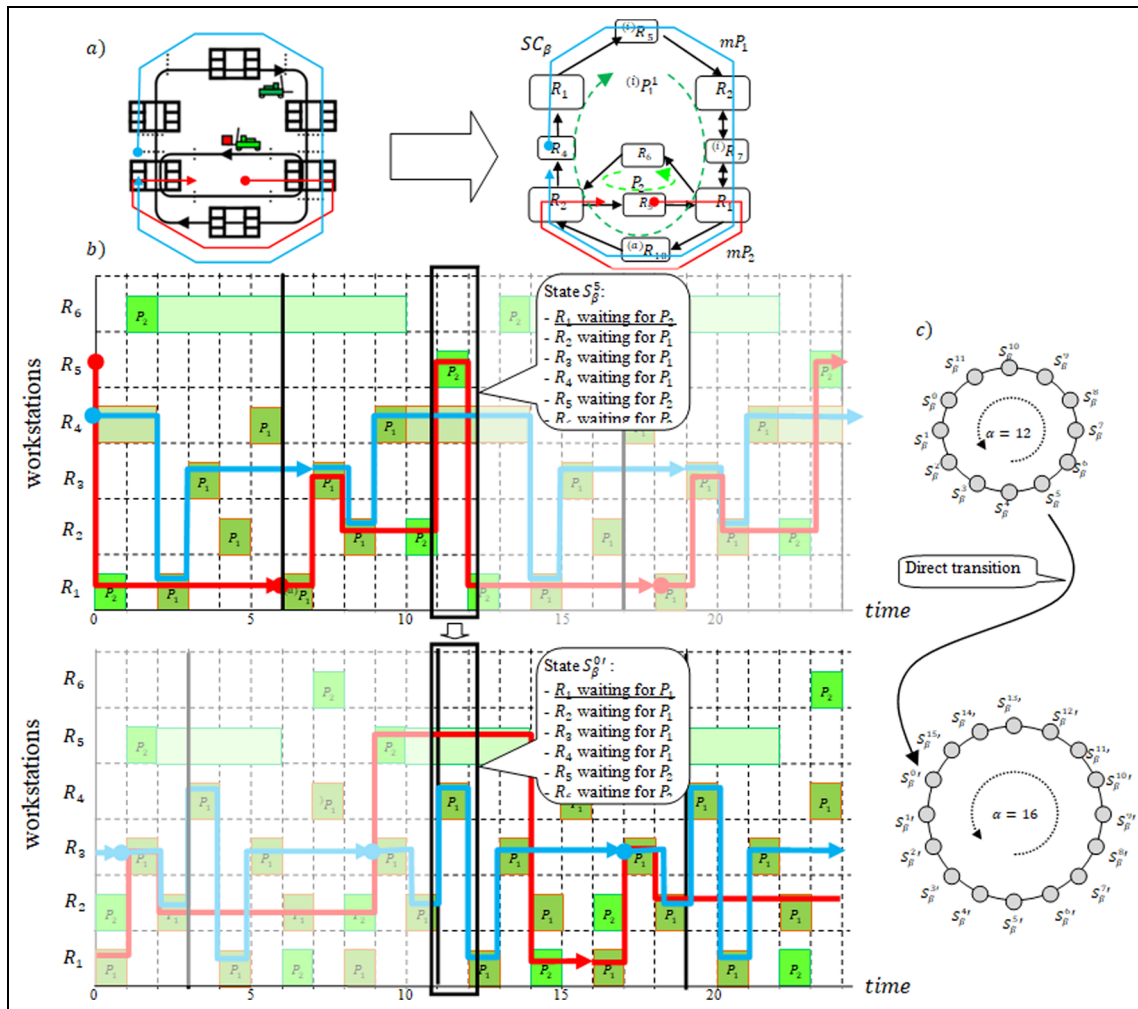


Figure 5. (a) Example of a flexible production job-shop and its SCCP model SC_β , (b) Gantt charts of cyclic schedules before and after disruption and (c) cyclic steady states corresponding to the Gantt charts.

S'_0 , in which the same processes are currently allocated to the same workstations. The transition occurs as a result of changing the order of processes queuing for access to workstation R_1 (from process P_2 to process P_1). In the example shown, the transition between states is direct, that is, there is no transient state.

The question that arises here concerns the possibility of returning from the newly reached cyclic steady state (which has arisen as a result of the disruption) to the previous one (directly preceding it). Unfortunately, in general, transitions of this sort are not always possible.⁴ In the absence of a transient state, the effects of a disruption cannot be eliminated, which means the system has no robustness to the occurrence of the disruption. To proceed, it is assumed that an SCCP is robust to given disruption Z if there is a state in the cyclic steady

$$x_{i,l} = \begin{cases} \max\{(x_{(a+1),b} + \Delta t), (x_{no_l,l} + t_{no_l,l} - \alpha)\} & \text{for } i = 1 \text{ and } a \in \{1 \dots no_b - 1\} \\ \max\{(x_{(a+1),b} + \Delta t), (x_{(i-1),l} + t_{(i-1),l})\} & \text{for } i = 2 \dots no_l \text{ and } a \in \{1 \dots no_b - 1\} \\ \max\{(x_{1,b} + \Delta t + \alpha), (x_{no_l,l} + t_{no_l,l} - \alpha)\} & \text{for } i = 1 \text{ and } a = no_b \\ \max\{(x_{1,b} + \Delta t + \alpha), (x_{(i-1),l} + t_{(i-1),l})\} & \text{for } i = 2 \dots no_l \text{ and } a = no_b \\ \max\{(x_{1,b} + \Delta t - \alpha), (x_{no_l,l} + t_{no_l,l} - \alpha)\} & \text{for } i = 1 \text{ \& if } o_{a,b} \text{ is executed in previous cycle} \\ \max\{(x_{1,b} + \Delta t - \alpha), (x_{(i-1),l} + t_{(i-1),l})\} & \text{for } i = 2 \dots no_l \text{ for } i = 1 \text{ \& if } o_{a,b} \text{ is executed in previous cycle} \end{cases} \quad (2)$$

states' space which, when the workstation operation times have been disrupted, makes it possible to restore the nominal production flow parameters (given by production period α and production takt time T). In other words, the ability of a system to accommodate a specific disruption Z is conditioned by the existence of a non-empty set of desired cyclic steady states. To assess the reachability of the desired cyclic state, one has to solve the following problem:

Sets:

- R : set of workstations, indexed by k
- P : set of local processes (e.g. transportation means, AGVs), indexed by l
- mP : set of multimodal processes (e.g. products) indexed by j
- Θ : set of dispatching rules, indexed by k

Constants:

- r : number of workstations
- n : number of local processes
- no_l : number of operations of process P_l
- m : number of multimodal processes
- nm_j : number of operations of multimodal process mP_j
- Δt : travel time between two successive workstations
- α^* : expected value of period of system
- T^* : expected value of production takt time
- $md_{i,j}$: execution time of i th workstation operation of process mP_j , determined by disruption Z

Decision variable:

- α : period of system
- T : production takt time
- $x_{i,l}$: starting time of i th operation of process P_l
- $t_{i,l}$: execution time of i th operation of process P_l
- $mx_{i,j}$: starting time of i th operation of process mP_j
- $mt_{i,j}$: execution time of i th transport operation of process mP_j

Constraints:

execution of operations $o_{i,l}$ of local processes P_l on non-shared workstation R_k ^{4,5}

$$x_{i,l} = \begin{cases} x_{no_l,l} + t_{no_l,l} - \alpha & \text{for } i = 1 \\ x_{(i-1),l} + t_{(i-1),l} & \text{for } i \in \{2 \dots no_l\} \end{cases} \quad (1)$$

execution of operations $o_{i,l}$ of local processes P_l on shared workstation R_k ^{4,5}

where $x_{a,b}$ is the starting time of operation $o_{a,b}$ of process P_b , executed on workstation R_k prior to operation $o_{i,l}$; $P_b \prec P_l|_{\sigma_k}$ – process P_b is executed (in conformity with priority dispatching rule σ_k) on workstation R_k before P_l .

execution of operations $mo_{i,j}$ of multimodal processes mP_j on workstation R_k

$$mx_{i,j} = x_{a,b} \quad \text{for each transportation sector} \quad (3)$$

$$mx_{i,j} = x_{c,d} + t_{c,d} - mt_{i,j} \quad \text{for input buffer} \quad (4)$$

$$md_{i,j} = \begin{cases} x_{c,d} + t_{c,d} - x_{a,b} & \text{for } x_{a,b} \leq x_{c,d} \\ x_{c,d} + t_{c,d} + \alpha - x_{a,b} & \text{for } x_{a,b} > x_{c,d} \end{cases} \quad (5)$$

$$mt_{i,j} = t_{a,b} \quad \text{for each transportation sector} \quad (6)$$

where $x_{a,b}$ is the starting time of operation of local process P_b which transfers process mP_j onto workstation R_k ; $x_{c,d}$ is the starting time of operation of local process P_d which transfers process mP_j from workstation R_k .

expected value of cyclic steady-state parameters

$$\alpha = \alpha^* \quad (7)$$

$$T = T^* \quad (8)$$

The goal of this problem is to find an answer to the following question:

- Do there exist values of the decision variables $t_{i,l}$, $mt_{i,j}$, $x_{i,l}$, $mx_{i,j}$, α , T , which, given the workstation operation times $md_{i,j}$ (determined by disruption Z), can guarantee the existence and reachability of a cyclic state (given by a cyclic schedule with given times $x_{i,l}$, $mx_{i,j}$) characterised by the expected production takt time T^* and period α^* ?

In other words, the solution is a system structure (specified by the values of transport operations $t_{i,l}$, $mt_{i,j}$) in which a cyclic state can be reached that makes it possible to correct the disruption that has occurred (i.e. a cyclic steady state that guarantees the achievement of the desired production takt time T^* and period α^*).

To find such a solution under the assumption that the transport operations of the processes are executed in a collision-free manner without deadlocks or starvation, that is, such that constraints (1)–(5) hold,^{4,5} is to find a solution to a relevant CSP.

Robust production flow schedule planning

The CSP

To solve the problem formulated in section ‘Cyclic steady states’ reachability’, it is enough to determine the values of the decision variables $t_{i,l}$, $mt_{i,j}$, $x_{i,l}$, $mx_{i,j}$,

α and T that satisfy constraints (1)–(8). In terms of constraint programming methods, such a problem is a CSP⁴ given by

$$CS = ((V, D), C) \quad (9)$$

where V is a set of decision variables $t_{i,l}$, $mt_{i,j}$, $x_{i,l}$, $mx_{i,j}$, α and T ; D is a family of domains of decision variables, where the values of the variables are integers, that is, $t_{i,l} \in \mathbb{N}^+$, $mt_{i,j} \in \mathbb{N}^+$, $x_{i,l} \in \mathbb{N}$, $mx_{i,j} \in \mathbb{N}$, $\alpha \in \mathbb{N}^+$ and $T \in \mathbb{N}^+$; C is a set of constraints (1)–(8).

For instance, in the SCCP shown in Figure 2, the representation of the CS problem contains 72 decision variables (characterizing six local processes $P_1 - P_6$ and one multimodal process mP_1) and 48 constraints. The decision variables for this system are shown in Figure 6, and the set of constraints relating these variables are given in Table 2.

A solution to CS problem (9) obtained for the new (changed) workstation operation times resulting from disruption Z allows one to determine whether there exists a cyclic steady state that can accommodate the disruption (allow for its correction).

Computational experiments

In order to evaluate the decidability of the problem of existence and reachability of cyclic steady states that allow for the correction of a disruption, we

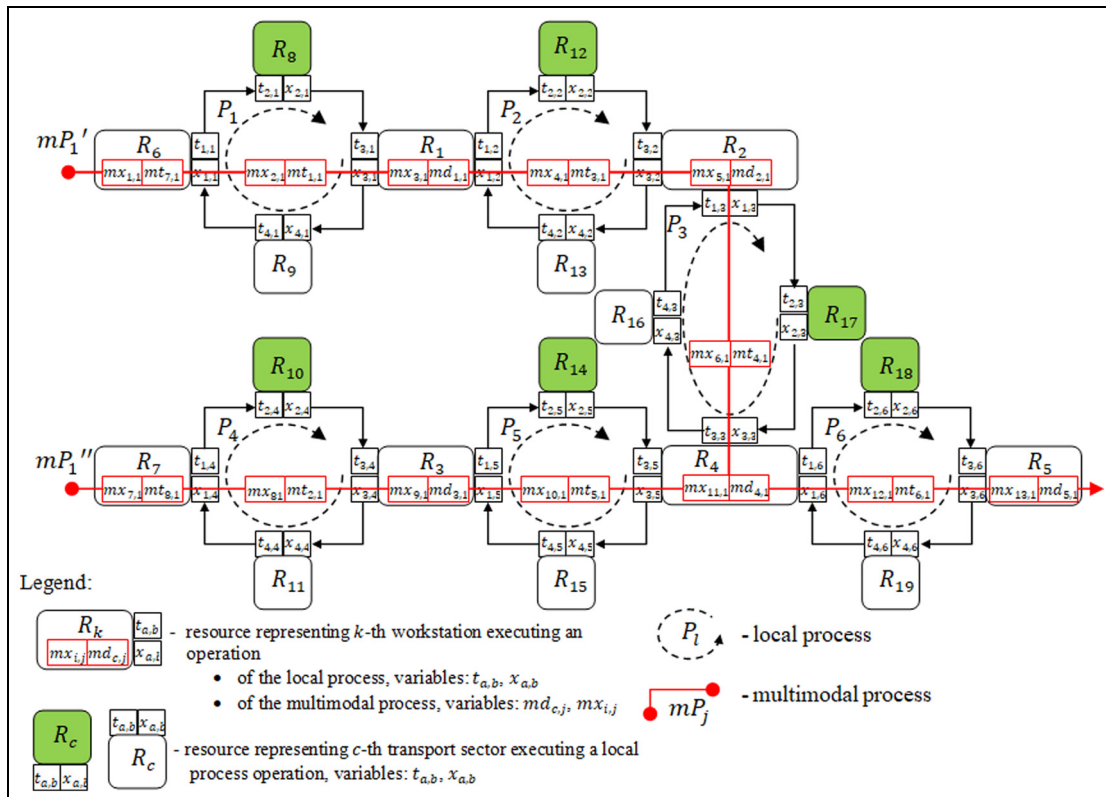


Figure 6. An SCCP featuring decision variables of problem (9).

Table 2. Constraints (1)–(8) for the system shown in Figure 6 ($\Delta t = 1$).

Constraints on execution of local processes			
C_1	$x_{1,1} = x_{4,1} + t_{4,1} - \alpha$	C_{25}	$x_{1,4} = x_{4,4} + t_{4,4} - \alpha$
C_2	$x_{2,1} = x_{1,1} + t_{1,1}$	C_{26}	$x_{2,4} = x_{1,4} + t_{1,4}$
C_3	$x_{3,1} = \max\{(x_{2,2} + 1), (x_{2,1} + t_{2,1})\}$	C_{27}	$x_{3,4} = \max\{(x_{2,5} + 1), (x_{2,4} + t_{2,4})\}$
C_4	$x_{4,1} = x_{3,1} + t_{3,1}$	C_{28}	$x_{4,4} = x_{3,4} + t_{3,4}$
C_5	$x_{1,2} = \max\{(x_{4,1} + 1 - \alpha), (x_{4,2} + t_{4,2} - \alpha)\}$	C_{29}	$x_{1,5} = \max\{(x_{4,4} + 1 - \alpha), (x_{4,5} + t_{4,5} - \alpha)\}$
C_6	$x_{2,2} = x_{1,2} + t_{1,2}$	C_{30}	$x_{2,5} = x_{1,5} + t_{1,5}$
C_7	$x_{3,2} = \max\{(x_{2,3} + 1), (x_{2,2} + t_{2,2})\}$	C_{31}	$x_{3,5} = \max\{(x_{2,6} + 1 - \alpha), (x_{2,5} + t_{2,5})\}$
C_8	$x_{4,2} = x_{3,2} + t_{3,2}$	C_{32}	$x_{4,5} = x_{3,5} + t_{3,5}$
C_9	$x_{1,3} = \max\{(x_{4,2} + 1 - \alpha), (x_{4,3} + t_{4,3} - \alpha)\}$	C_{33}	$x_{1,6} = \max\{(x_{4,3} + 1), (x_{4,6} + t_{4,6} - \alpha)\}$
C_{10}	$x_{2,3} = x_{1,3} + t_{1,3}$	C_{34}	$x_{2,6} = x_{1,6} + t_{1,6}$
C_{11}	$x_{3,3} = \max\{(x_{4,5} + 1), (x_{2,3} + t_{2,3})\}$	C_{35}	$x_{3,6} = x_{2,6} + t_{2,6}$
C_{12}	$x_{4,3} = x_{3,3} + t_{3,3}$	C_{36}	$x_{4,6} = x_{3,6} + t_{3,6}$
Constraints on execution of multimodal processes			
C_{13}	$md_{1,1} = x_{1,2} + t_{1,2} + \alpha - x_{3,1}$	C_{37}	$md_{2,1} = x_{1,3} + t_{1,3} + \alpha - x_{3,2}$
C_{14}	$md_{3,1} = x_{1,5} + t_{1,5} + \alpha - x_{3,4}$	C_{38}	$md_{4,1} = x_{1,6} + t_{1,6} + \alpha - x_{3,3}$
C_{15}	$mt_{1,1} = t_{2,1}$	C_{39}	$mt_{2,1} = t_{2,4}$
C_{16}	$mt_{3,1} = t_{2,2}$	C_{40}	$mt_{4,1} = t_{2,3}$
C_{17}	$mt_{5,1} = t_{2,5}$	C_{41}	$mt_{6,1} = t_{2,6}$
C_{18}	$mx_{1,1} = x_{1,1} + t_{1,1} - mt_{7,1}$	C_{42}	$mx_{2,1} = x_{2,1}$
C_{19}	$mx_{3,1} = x_{3,1}$	C_{43}	$mx_{4,1} = x_{2,2}$
C_{20}	$mx_{5,1} = x_{3,2}$	C_{44}	$mx_{6,1} = x_{2,3}$
C_{21}	$mx_{7,1} = x_{1,4} + t_{1,4} - mt_{8,1}$	C_{45}	$mx_{8,1} = x_{2,4}$
C_{22}	$mx_{9,1} = x_{3,4}$	C_{46}	$mx_{10,1} = x_{2,5}$
C_{23}	$mx_{11,1} = x_{3,5}$	C_{47}	$mx_{12,1} = x_{2,6}$
C_{24}	$mx_{13,1} = x_{3,1}$	C_{48}	$T = c \cdot \alpha, c \in \mathbb{N}^+$

implemented CSP (9) in the Oz Mozart constraint programming environment (Windows 10, Intel Core Duo2 3.00 GHz, 4 GB RAM). Three types of disruptions are considered. They concern changes in workstation operation times in the SCCP of Figure 6, for which there occurs the following:

A change in the location of the system's bottleneck (a change in production flow) resulting in reduced production takt time T (disruption Z_1);


No change in the system's bottleneck and no change in production takt time T (disruption Z_2);

A change in the system's bottleneck resulting in increased production takt time T (disruption Z_3).

The experiments were carried out under the assumption that both workstation operation times (determined by the type of disruption) and transport operation times (which were adjusted to correct the disruption) fell within the intervals specified in Table 1. In addition, shop floor transport cycle times $\Delta t = 1$ and $t_{i,l} = 1$ were assumed for workstation ($R_1 - R_6$) and 'return transport' ($R_9, R_{11}, R_{13}, R_{15}, R_{16}, R_{19}$).

Disruption type Z_1 . In the system of Figure 6, operating in the nominal cyclic steady state of production flow (with nominal workstation operation times as given in

Table 1) – see Figure 3 – operations on workstations R_1, R_3 and R_4 get disrupted.

Figure 7 shows a schedule generated as a result of the disruption of the nominal state. Cycles 1 and 2 represent the nominal state schedule for which the production takt time is $T = 11$. In cycles 3 and 4, a disruption, marked as , occurs at workstations R_1, R_3 and R_4 . As a result of the disruption, the operation times change from the nominal values 6, 5 and 8 to 7, 4 and 6, respectively. The reduction in operation time on workstation R_4 (from 8 to 6) causes a change in the bottleneck load, which in turn results in a new cyclic steady state of the SCCP (cycles 3–5 of the schedule in Figure 7) with production takt time $T = 9$.

In order to assess the possibility of correcting this type of disruption, CS problem (9) was solved. The first acceptable solution was obtained in less than 1 s. The solution provided transport operation times $mt_{i,j}$ (see Table 3) and a cyclic steady state with production takt time $T = 11$. The schedule for this cyclic steady state is illustrated in cycles 10 and 11, and so on, of Figure 8.

A transition to the newly obtained cyclic steady state is possible through a transient state (Figures 7 and 8). The transition occurs in two steps. The first one involves reaching a cyclic steady state with the desired production takt time. To allow the system to reach this state, the times of transport operations at workstations R_8, R_{10}, R_{17} and R_{18} were corrected to match the values shown in Table 3 (times $mt_{1,1}, mt_{2,1}, mt_{4,1}, mt_{6,1}$) – the

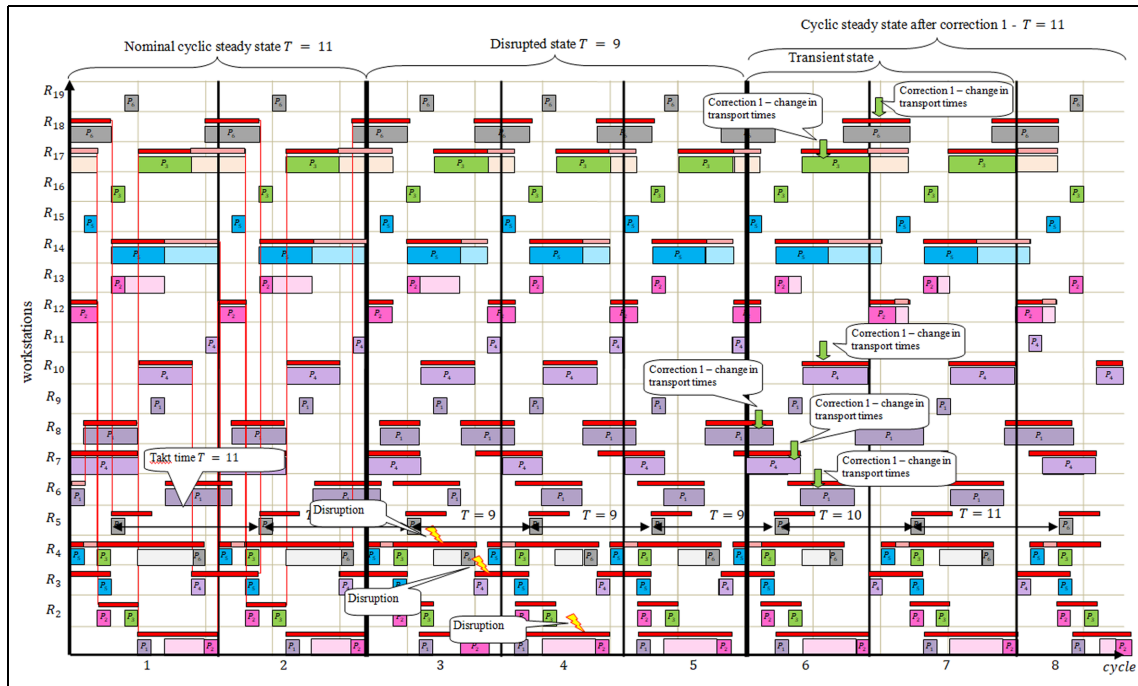


Figure 7. Gantt's chart of cyclic schedule for the SCCP of Figure 6 affected by disruption type Z_1 .

Table 3. Transport times allowing for correction of disruption Z_1 .

	Nominal-state times	Disruption Z_1		Nominal-state times	Correction
$md_{1,1}$	6	7	$mt_{1,1}$	4	5
$md_{2,1}$	3	3	$mt_{2,1}$	4	5
$md_{3,1}$	5	4	$mt_{3,1}$	2	2
$md_{4,1}$	8	6	$mt_{4,1}$	4	5
$md_{5,1}$	3	3	$mt_{5,1}$	4	4
			$mt_{6,1}$	4	5
			$mt_{7,1}$	5	5
			$mt_{8,1}$	5	5

operations with changed transportation times are marked in Figures 7 and 8 with green arrows \downarrow . The changes introduced resulted in a transition to a new cyclic steady state with production takt time $T = 11$ (cycles 7 and 8). Note that the schedule corresponding to the newly obtained cyclic steady state is shifted in time relative to the nominal schedule by 7 time units (see Figure 8).

This means that the cyclic steady state obtained is characterized by the same production rate as before the disruption, but it does not allow to meet the same (pre-scheduled) production completion dates. Newly manufactured products are ready to be picked up 7 units earlier than the pre-scheduled production completion date. The second step of the transition focuses on correcting the time shift. To accommodate the time shift, execution of operations on workstations R_1 , R_7 , R_{13} , R_{14} , R_{16} and R_{19} was suspended for one cycle (cycle

9; Figure 7). The new schedule thus obtained guarantees the same production parameters (production takt time T and timely delivery) as before the disruption.

The process of correcting the disruption described by the line graph in Figure 9(a) is a function which assigns production completion times to the successive production cycles. According to this graph, in the nominal state (blue line), finished products leave the production line at t.u. 11, 22, 33, 44, ... The red line represents the disrupted-and-corrected production flow (and the associated production completion times). The correction time spans six production cycles (cycles 3–9). The transitions between reachable cyclic steady states corresponding to the successive correction steps are shown in Figure 9(b).

Disruption type Z_2 . In the next experiment, it is assumed that the change in the operation times takes place on

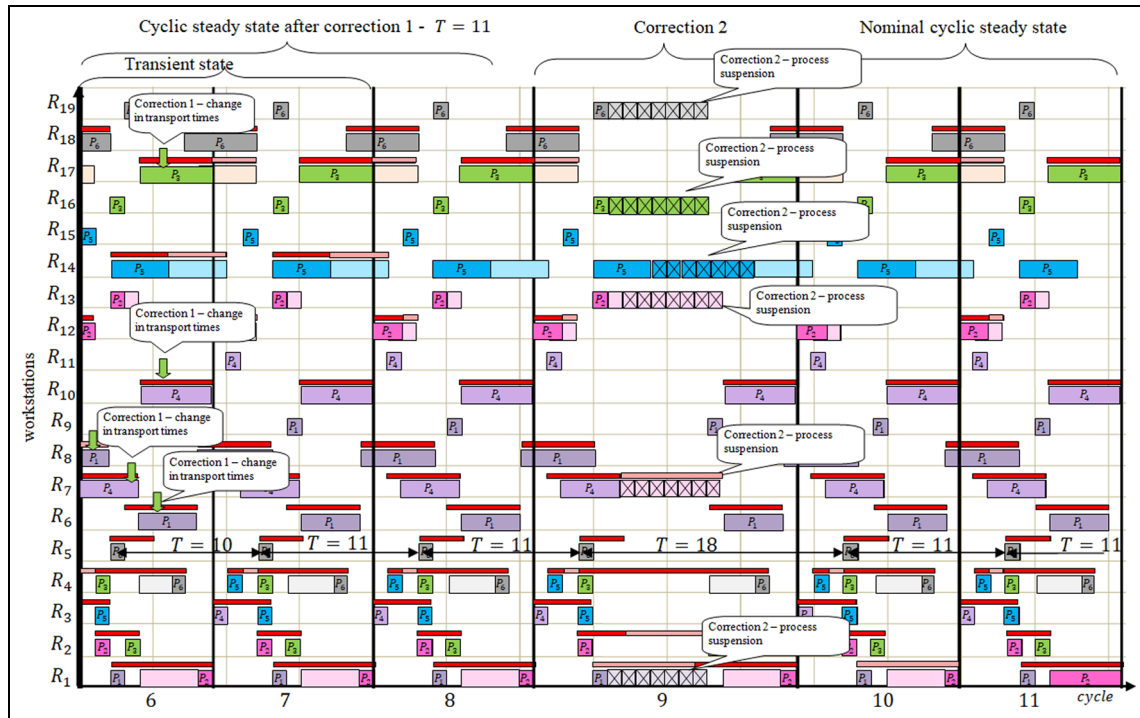


Figure 8. A Gantt's chart showing the correction made to the schedule of Figure 7, restoring the previously specified production completion dates.

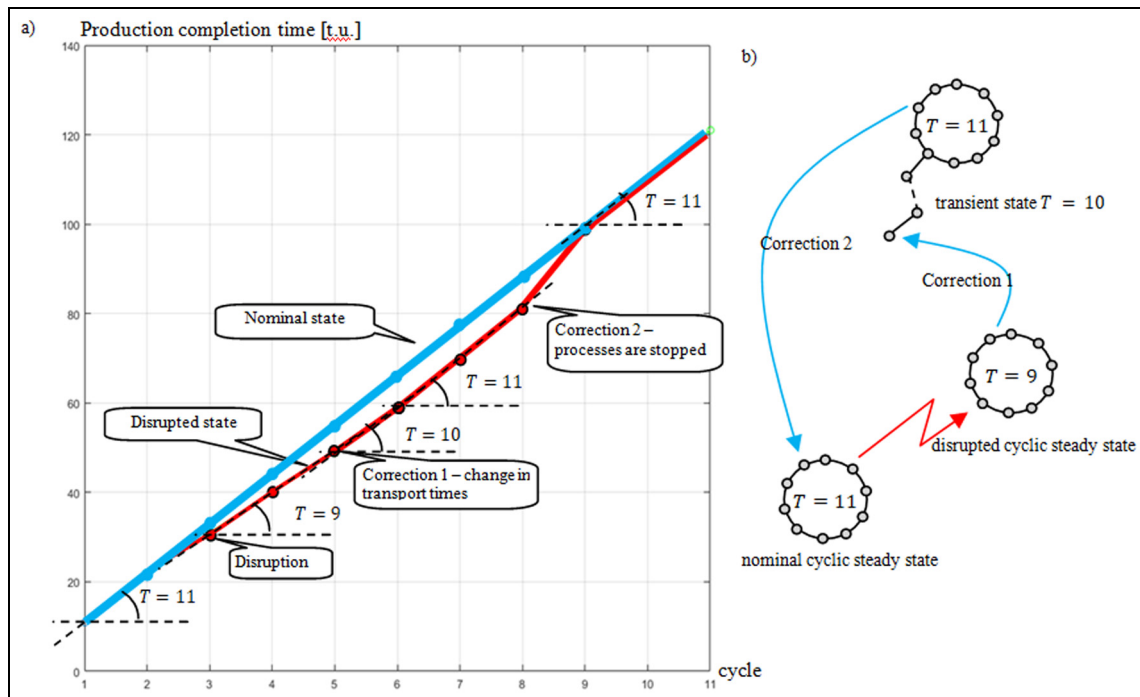


Figure 9. (a) Production completion charts for nominal cyclic steady state (blue line) and disrupted state (red line) and (b) the corresponding changes in cyclic steady states.

workstations R_1 , R_2 and R_3 . It is also assumed that these operation times are increased from the nominal values 6, 3 and 5 to 7, 4 and 6, respectively. The schedule

corresponding to the newly reached cyclic steady state is shown in Figure 10 (the disturbance occurs in cycle 3). It should be noted that the changes in the times of

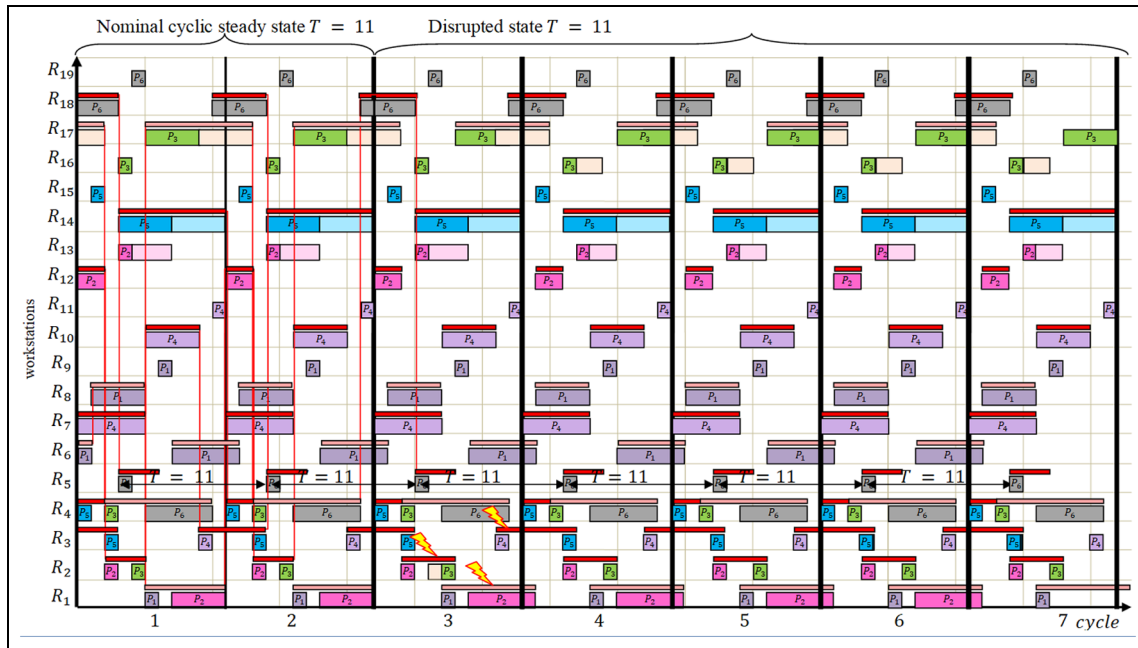


Figure 10. Gantt's chart of cyclic schedule for the SCCP of Figure 6 affected by disruption type Z_2 .

execution of operations do not affect the production takt time. In the newly reached cyclic steady state, the production takt time is still $T = 11$. The fact that there is no change in the production flow is a consequence of there being no change in the location of the system's bottleneck. The bottleneck is still workstation R_4 and its process time has not changed.

This means that adjustment in the situation under consideration is not necessary – the production completion graphs for the nominal and the disrupted states overlap (Figure 11). This case shows that not every change in operation times requires a correction. Such situations occur when the disruption does not affect the bottleneck of the system (i.e. it does not change the location of the bottleneck or the pre-scheduled production takt time).

Disruption type Z_3 . In another experiment, a disruption is considered which involves an increase in operation time on only one workstation R_4 – the operation time increases from the nominal value of 8 to 9. Since workstation R_4 is still the system's bottleneck, the increase in operation time on this workstation leads to an increase in its load and, consequently, to lengthening of the production takt time. The cyclic steady state obtained is executed with production takt time $T = 12$. The schedule for this state is shown in Figure 12 (the disruption occurs in cycle 3).

Just as in the case of disruption type Z_1 , an attempt was made to determine the structure parameters (times of transport operations $mt_{1,1} - mt_{8,1}$) which would allow the system to reach a new cyclic steady state that

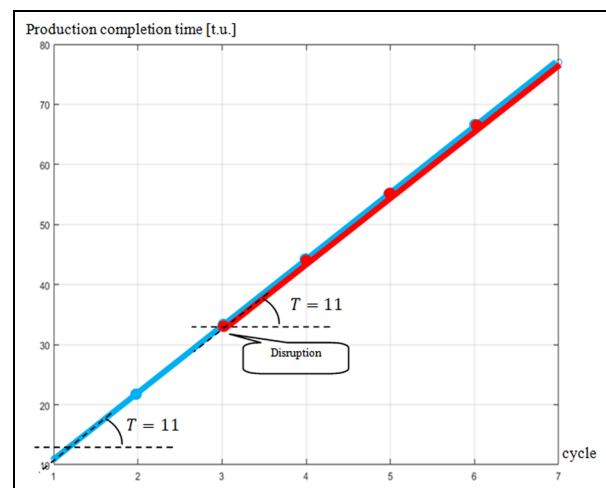


Figure 11. Production completion charts for nominal cyclic steady state (blue line) and the disrupted state – disruption Z_3 (red line).

would accommodate (correct) the disruption in question. For this purpose, CS problem (9) was solved (in 15s), for which, however, no acceptable solution was found. The lack of a solution means that in the space of admissible steady states (determined by the adopted time intervals – Table 1), there is no cyclic steady state that makes it possible to correct the disruption.

The reason for this is that changes in transport operation times in this case do not lead to changes in the times of operations executed at the bottleneck. The workstation whose operation has been disrupted remains the bottleneck of the system which determines the production takt time. This means that if a

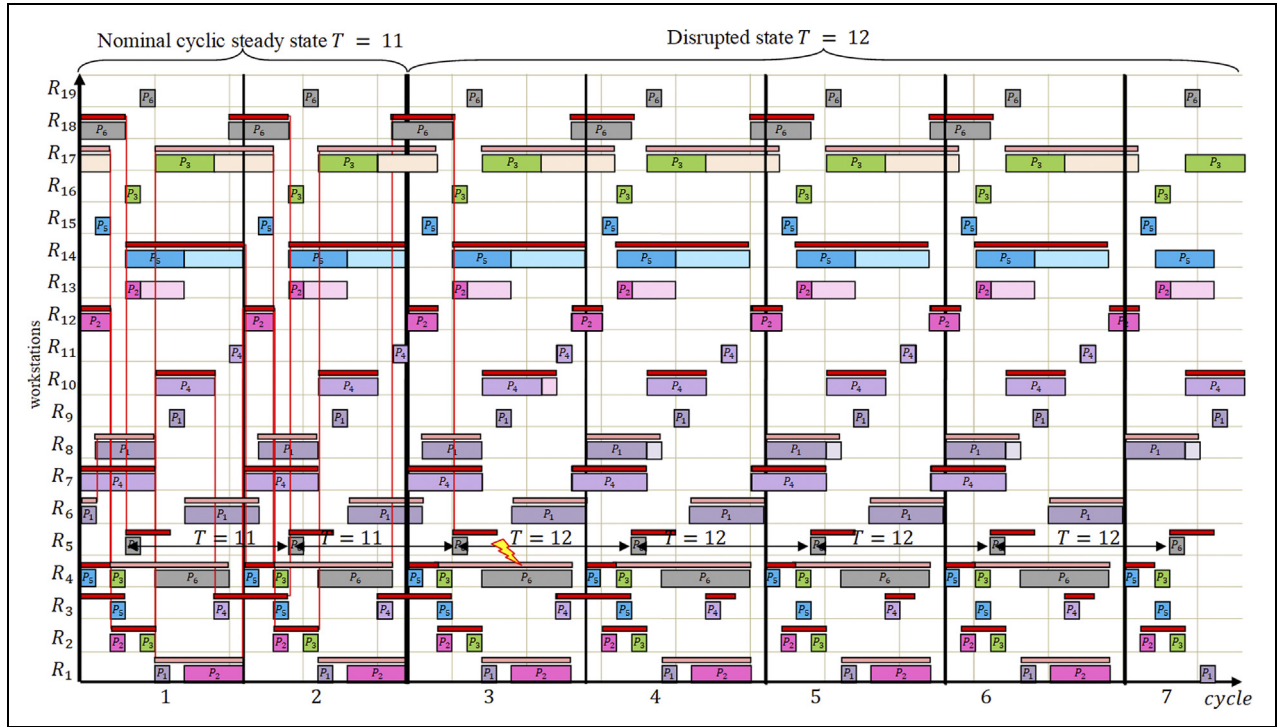


Figure 12. Gantt's chart of a cyclic schedule for the SCCP of Figure 6 affected by disruption type Z_3 .

disturbance occurs solely in the system's bottleneck, a return to the nominal state is only possible when this disruption is removed. Figure 12 shows Gantt's charts that illustrate differences in production takt time between the disrupted state and the nominal state. Figure 13(a) shows production completion charts for nominal cyclic steady state and disrupted state, and Figure 13(b) presents the corresponding change in cyclic steady states.

The condition for correcting disruption Z

The experiments show that disruption Z is not correctable for all systems in all situations. To assess whether a correction is feasible, it is necessary to determine transport operation times which make it possible for a given disruption to be corrected (i.e. allow a system to reach a cyclic steady-state characterized by the same production takt time T and period α as the nominal state). Any such assessment requires finding a solution to a relevant CSP CS (9). As the third experiment shows, such a solution does not always exist – in such cases the space of admissible solutions to the problem is empty. In general, to determine that a specific problem CS does not have a solution, one has to invest considerable computational effort in searching the whole space of potential solutions – in other words, a problem like this is NP-hard. In practice, the scale of real-life problems limited by implemented constraint programming methods does not exceed an amount of 30 simultaneously executed processes.³¹

However, in case of SCCP systems possessing a regular topology where a method based on the principle of composition of multiple elementary transport networks can be used,⁶ the size of scheduling problems can be dramatically extended even to around 200 processes. In general, however for the large-scale systems normally encountered in everyday practice, this type of assessment may simply be unworkable. Therefore, it seems necessary to find conditions enabling a preliminary assessment of the existence of an admissible solution (decidability of the problem of correcting a given disruption).

The experiments performed in this study indicate that correction of a disturbance is possible only in cases where the disturbance does not interfere with the location and load of the bottleneck (does not lead to a load reduction). This observation allows a formulation of the following condition:

If a disruption in workstation operation time occurring on the workstation of a SCCP that is the bottleneck of this SCCP results in an increase in production takt time T , then a cyclic steady state that allows for appropriate correction of the disruption is not reachable in this system.

The condition presented here is one of the possible sufficient conditions, that is, a condition which, when satisfied, implies that there is no solution to the CS problem and consequently that the effects of the disruption are impossible to correct. Hence, verification of

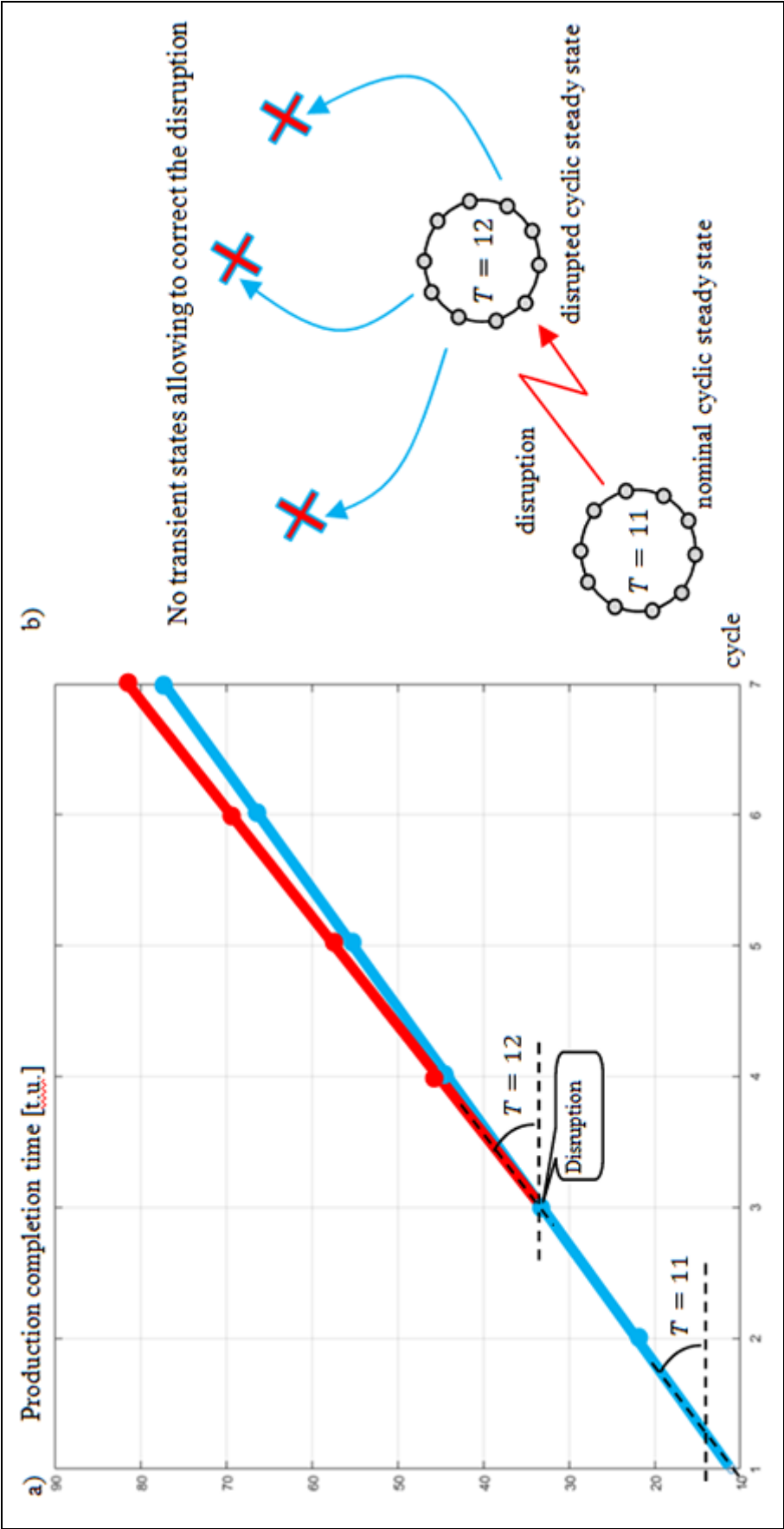


Figure 13. (a) Production completion charts for nominal cyclic steady state (blue line) and disrupted state (red line) and (b) the corresponding change in cyclic states.

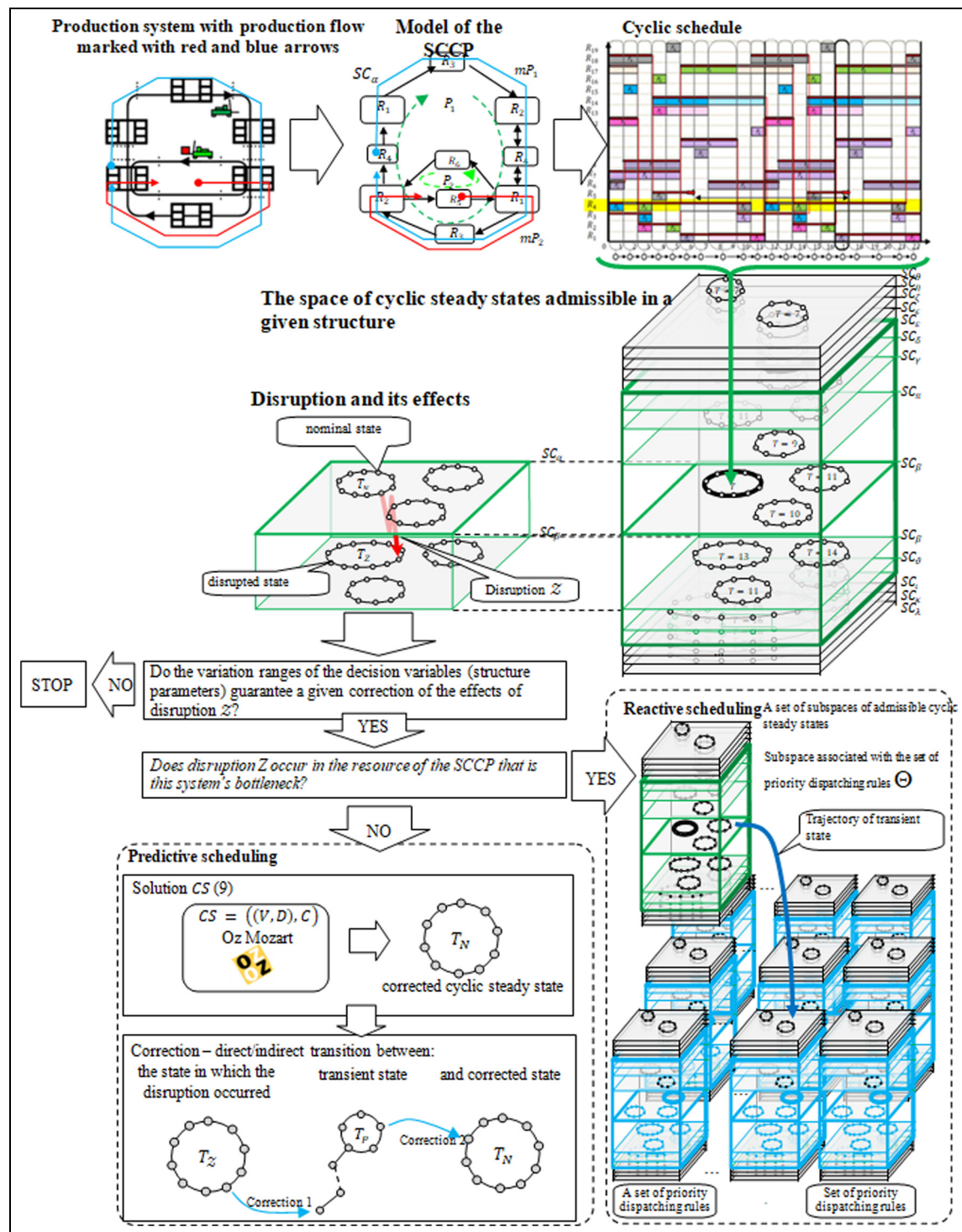


Figure 14. A schematic diagram of the research methodology used.

this condition allows one to determine (avoiding unnecessary calculations) the robustness of the cyclic schedule under consideration. The place and importance of this condition in the research methodology adopted in this study are illustrated in Figure 14. Depending on whether the condition is satisfied or not, one of two alternative procedures is selected. A first procedure (the

one presented in this article) consists in removing the effects of disruptions by making appropriate corrections to operation times, that is, by introducing changes resulting in the transition between selected cyclic steady states within the subspace of admissible solutions parameterized by the same set of priority dispatching rules. A second procedure essentially involves searching

subspaces parameterized by different sets of priority dispatching rules for such cyclic steady states reachable in those subspaces which give rise to specific changes in the values of production flow parameters, for example, minimum changes relative to the nominal values.

In summary, unlike the solutions proposed in the literature, in which each failure automatically initiates a correcting procedure (e.g. approaches based on max-plus algebra¹⁹), the solution advanced in this article is that corrective action is only taken when the disruption occurs in locations other than the system's bottleneck. Our solution is limited to cases of individually occurring disruptions of a production system, in particular cases of permanent disruptions. This means that this avenue of research should be extended to also cover cases of individual but short-lasting disruptions. Investigations of such cases emerging in a given cyclic steady states' space should focus on formulation of appropriate sufficient conditions. Fulfilment of such conditions should guarantee both a transition to an alternative steady state which removes the effects of a given disruption and a return to the steady state that preceded the occurrence of the disruption. A separate study should be conducted to examine cases of simultaneous occurrence of several types of disruptions. A particularly important issue that remains open in this respect is the question of decidability of existence and reachability of alternative cyclic steady states that guarantee the functioning of a system under component failure conditions.

Concluding remarks

The proposed approach was compared in terms of scheduling efficiency with a method based on unified max-plus algebra and a model predictive control framework.¹⁹ The results show that, for most disruptions causing variation in operation times of workstations which are not system bottlenecks, the two approaches have similar levels of re-scheduling capacity. However, in the case of disruptions to bottleneck workstations, the new approach based on the state space reachability framework is superior as it eliminates unnecessary searching for schedules that do not exist.

The proposed declarative modelling-based methodology, which seems to be an attractive alternative to other methodologies driven by mathematical programming, computer simulation and/or OR-based metaheuristics, allows us to formulate the problems of cyclic scheduling in flexible manufacturing stated either in direct (aimed at analysis) or reverse (aimed at synthesis) ways. So the main advantage of the proposed approach follows from the fact that any of the considered decision variables can be seen as indicators of the system's

performance, and none of them requires that relevant goal functions are set up.

The declarative character of the model developed in this study makes it well suited for various types of manufacturing situations encountered in practice: the words machine, job and machine shop are only used in a symbolic sense. Examples of systems to which the model can be applied include urban transport systems and data-, energy- and goods-transmission networks. In all these applications, a dominant role is played by the problems of crisis management and the related issues of reliability of the proposed structural and organizational solutions.

It is also worth noting that in addition to the prototyping of production flow organization and planning (scheduling) methods, one can also take an alternative approach in which manufacturing technologies are changed/selected to organize production flow in a way that will allow it to meet ad hoc the customers'/recipients' changing requirements. New research in that direction would involve determining the conditions allowing a company to re-schedule cyclic production in conformity with the changes (disruptions) imposed by the customers' new requirements. These changes may concern both the diversity of the product range and production volume. They may also include changes in delivery dates and changes in the size of the batches to be picked up by the customer. The possibility of customers having these types of requirements and expectations makes it necessary to reconsider the familiar questions of whether the available structural and organizational potential of the company guarantees its proper functioning (profitability) given the changes in customer expectations, and what structural and organizational changes will the company have to make to ensure its competitiveness considering that it has access to new technologies and/or that there are changes in sales principles? The application of the proposed state reachability space approach in the context of the questions raised above will also be a focus of our future research.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

References

1. Levner E, Kats V and De Pablo DAL. Cyclic scheduling in robotic cells: an extension of basic models in machine scheduling theory. In: Levner E (ed.) *Multiprocessor*

- scheduling: theory and applications. Aarhus: I-Tech Education and Publishing, 2007, pp.1–20.
2. Bocewicz G and Banaszak Z. Scheduling for multi-modal cyclic transport systems. *Pomiary Autom Robot* 2012; 2: 106–113.
 3. Bocewicz G, Wójcik R, Banaszak ZA, et al. Multimodal processes rescheduling: cyclic steady states space approach. *Math Probl Eng* 2013; 2013: 407096.
 4. Bocewicz G, Muszyński W and Banaszak Z. Models of multimodal networks and transport processes. *B Pol Acad Sci: Tech* 2015; 63: 635–650.
 5. Bocewicz G, Nielsen IE and Banaszak ZA. Production flows scheduling subject to fuzzy processing time constraints. *Int J Comp Integ M* 2016; 29: 1105–1127.
 6. Bocewicz G, Banaszak Z and Nielsen I. Multimodal processes prototyping subject to grid-like network and fuzzy operation time constraints. *Ann Oper Res*. Epub ahead of print 26 May 2017. DOI: 10.1007/s10479-017-2468-5.
 7. Polak M, Majdzik P, Banaszak Z, et al. The performance evaluation tool for automated prototyping of concurrent cyclic processes. *Fund Inform* 2004; 60: 269–289.
 8. Bożejko W, Uchroński M and Wodecki M. Block approach to the cyclic flow shop scheduling. *Comput Ind Eng* 2015; 81: 158–166.
 9. Błażewicz J, Ecker K, Pesch E, et al. *Handbook on scheduling: from theory to applications*. New York: Springer, 2007, p.647.
 10. Banaszak Z (ed.). *Modeling and control of FMS: Petri net approach*. Wrocław: Technical University Press, 1991, p.323.
 11. Pascal D. *Lean production simplified*. 2nd ed. New York: CRC Press, 2007, p.192.
 12. Levner E, Kats V, De Pablo DAL, et al. Complexity of cyclic scheduling problems: a state-of-the-art survey. *Comput Ind Eng* 2010; 59: 352–3618.
 13. Li Z, Janardhanan MN, Tang Q, et al. Co-evolutionary particle swarm optimization algorithm for two-sided robotic assembly line balancing problem. *Adv Mech Eng* 2016; 89: 1–14.
 14. Kampmeyer T. *Cyclic scheduling problems*. PhD Thesis, Fachbereich Mathematik/Informatik Universit, Osnabrück, 2006.
 15. Kubale M and Nadolski A. Chromatic scheduling in a cyclic open shop. *Eur J Oper Res* 2005; 1643: 585–591.
 16. Munier A. The basic cyclic scheduling problem with linear precedence constraints. *Discrete Appl Math* 1996; 643: 219–238.
 17. Zhang L, Li X, Gao L, et al. Predictive/reactive scheduling with uncertain disruptions. In: *Proceedings of the 41st international conference on computers & industrial engineering*, Los Angeles, CA, 23–25 October 2011, pp.260–265. Computers & Industrial Engineering.
 18. Katragini K, Vallada E and Ruiz R. Flow shop rescheduling under different types of disruption. *Int J Prod Res* 2013; 51: 780–797.
 19. Seybold L, Witeczak M, Majdzik P, et al. Towards robust predictive fault-tolerant control for a battery assembly system. *Int J Appl Math Comput Sci* 2015; 25: 849–862.
 20. El-Najdawi MK. Multi-cyclic flow shop scheduling: an application in multi-stage, multiproduct production processes. *Int J Prod Res* 1997; 35: 3323–3332.
 21. Dileep G, Sumer D and Aggarwal C. Multi-machine, multi-product production scheduling and inventory control. *Decision Sci* 1978; 9: 577–595.
 22. Francis P and Smilowitz K. Modeling techniques for periodic vehicle routing problems. *Transport Res B: Meth* 2006; 40: 872–884.
 23. Coene S and Arnout A. On a periodic vehicle routing problem. *J Oper Res Soc* 2010; 61: 1719–1728.
 24. Ullrich CA. Integrated machine scheduling and vehicle routing with time windows. *Eur J Oper Res* 2013; 227: 152–165.
 25. Haghani A and Oh SC. Formulation and solution of a multi-commodity, multi-modal network flow model for disaster relief operations. *Transp Res A: Pol* 1996; 30: 231–250.
 26. Azi N, Gendreau M and Potvin JY. A dynamic vehicle routing problem with multiple delivery routes. *Ann Oper Res* 2012; 199: 103–112.
 27. Rousseau LM, Gendreau M and Pesant G. Using constraint-based operators to solve the vehicle routing problem with time windows. *J Heuristics* 2002; 8: 43–58.
 28. Kilby P and Urli T. Fleet design optimisation from historical data using constraint programming and large neighbourhood search. *Constraints* 2016; 211: 2–21.
 29. Reed M, Yiannakou A and Evering R. An ant colony algorithm for the multi-compartment vehicle routing problem. *Appl Soft Comput* 2014; 15: 169–176.
 30. Bocewicz G, Klempous R and Banaszak Z. Declarative approach to DSS design for supervisory control of production orders portfolio. In: *Proceedings of 7th IEEE international conference on cognitive informatics (CogInfoCom)*, Wrocław, 16–18 October 2016, pp.385–390. New York: IEEE.
 31. Bocewicz G, Nielsen I and Banaszak Z. Iterative multi-modal processes scheduling. *Ann Rev Control* 2014; 38: 113–132.
 32. Ageeva Y. *Approaches to incorporating robustness into airline scheduling*. Master of Engineering in Computer Science, Massachusetts Institute Of Technology, 2000.
 33. Herroelen W and Leus R. Robust and reactive project scheduling: a review and classification of procedures. *Int J Prod Res* 2004; 42: 1599–1620.
 34. Fahmy SA, Balakrishnan S and ElMekawy TY. A generic deadlock-free reactive scheduling approach. *Int J Prod Res* 2009; 47: 5657–5676.
 35. Sahner R, Trivedi K and Puliafito A. *Performance and reliability analysis of computer systems: an example-based approach using the sharpe software package*. New York: Springer, 2012, p.404.
 36. Abrams M, Doraswamy N, Mathur A, et al. Visual analysis of parallel and distributed programs in the time, event, and frequency domains. *IEEE T Parall Distri* 1992; 3: 672–685.
 37. Yan F, Dridi M and El Moudni A. An autonomous vehicle sequencing problem at intersections: a genetic algorithm approach. *Int J Appl Math Comp* 2013; 231: 183–200.
 38. Houssin L. Cyclic job-shop problem and (max, plus) algebra. In: *Proceedings of world IFAC congress (IFAC 2011)*, Milan, 28 August–2 September 2011, pp.2717–2721, <https://hal.archives-ouvertes.fr/hal-00667948/document>