

Available online at www.sciencedirect.com

# **ScienceDirect**

journal homepage: http://www.elsevier.com/locate/acme

# **Original Research Article**

# **Optimization of production process for resource utilization**



# Wojciech Bozejko<sup>a</sup>, Anna Burduk<sup>b,\*</sup>, Jaroslaw Pempera<sup>a</sup>, Mieczyslaw Wodecki<sup>c</sup>

<sup>a</sup> Wroclaw University of Science and Technology, Faculty of Electronics, Department of Control Systems and Mechatronics, Janiszewskiego 11-17, 50-372 Wroclaw, Poland

<sup>b</sup> Wroclaw University of Science and Technology, Centre for Advanced Manufacturing Technologies (CAMT), Faculty of Mechanical Engineering, Lukasiewicza Street 5, 50-370 Wroclaw, Poland

<sup>c</sup> Wroclaw University of Science and Technology, Faculty of Electronics, Telecommunications and Teleinformatics Department, Janiszewskiego 11-17, 50-372 Wroclaw, Poland

#### ARTICLE INFO

Article history: Received 14 December 2018 Received in revised form 3 June 2019 Accepted 4 July 2019 Available online

Keywords: Optimization Production cell No-store constraint Cyclic scheduling

#### ABSTRACT

The paper presents a method of scheduling execution of technological operations in work centres in a company manufacturing seats for passenger cars. To that end, an optimization algorithm based on the B&B method was developed, aimed at minimizing the number of employees necessary for operation of the work centres and at maximizing capacity of the manufacturing process. Efficiency and effectiveness of the algorithm was verified on the grounds of real data coming from the manufacturing company.

© 2019 Politechnika Wroclawska. Published by Elsevier B.V. All rights reserved.

## 1. Introduction

The majority of optimization problems in the theory of scheduling, dedicated to manufacturing systems, is formulated for non-cyclic systems (batch scheduling), for the given a priori set of tasks to be realized. Differently, production systems, which realize multi-assortment production, in large quantities, at the slow-changing in time range of products, manufacture products only in a cyclic manner [2,4,6]. Moreover, modern systems of this type satisfy market demand by a homogenous in time production (whose range varies according to market demand) or by regular supply of the right mix of assortments output (see e.g. Ref. [18]). Assuming the technical feasibility of both the above mentioned solutions, the latter approach is economically more attractive because it eliminates or reduces the size of the finished goods warehouse. Optimization of such a system usually boils down to

\* Corresponding author.

https://doi.org/10.1016/j.acme.2019.07.002

E-mail addresses: wojciech.bozejko@pwr.edu.pl (W. Bozejko), anna.burduk@pwr.edu.pl (A. Burduk), jaroslaw.pempera@pwr.edu.pl (J. Pempera), mieczyslaw.wodecki@pwr.edu.pl (M. Wodecki).

<sup>1644-9665/© 2019</sup> Politechnika Wroclawska. Published by Elsevier B.V. All rights reserved.

minimizing the period of the cycle, for a fixed mixture of tasks (products) in the cycle, resulting in improved system performance and improvement in machine utilization. This is especially important in mass production when producing in large quantities of identical or similar products. Then, storage costs can account for a significant part of the cost of producing the final product. In this type of companies, "just in time" or "only in sequence" strategies are used [1].

Usually, further improvement of the system efficiency is achieved by eliminating or reducing of intermediate storage of prefabrications in warehouses or between their intermediate positions in factories, these conditions are known in the literature as 'no store', 'a limited store', 'buffer', 'waiting time constraints'. Cyclic problems belong to a unique, littleresearched subclass of scheduling problems because they concern a non-regular criterion. Cyclic systems (including the additional constraints of storage) are becoming more and more popular among both practitioners and theorists. They are object of interest to the researchers mainly because of their strong practical importance and difficulty in obtaining adequately efficient algorithms for solving problems [12], also with using a graph modelling [14].

In the enterprise at work the process of production of seats and other car equipment includes product design, computer and real-life tests related to the safety of their use and the production of finished products. Extensive experience in both designing and production of products allows the company to produce many products for many contractors [7].

Despite the significant and continuous development of robots and automation devices, the most important element of the production system is man. In advanced production systems, its role is often the precise placement of processed products in the holders of automatic welding modules, making welding corrections or welding in hard-to-reach places, carrying out product quality control and levelling and polishing surfaces at various stages of production.

Nowadays, manufacturing companies more and more struggle with a shortage of qualified employees. Financial inducements increase production costs. Therefore, effective human resource management becomes more and more



Fig. 1 - Whole production process flow.

necessary. In this situation, a good solution can be application of smart methods for building production schedules that make it possible, on the one hand, to minimize execution time of production orders and, on the other hand, to use in the process the possibly smallest number of highly qualified employees.

# 2. Research performed in industry

The research was carried out in a company that produces components for the automotive industry. Due to the nature of the elements, most of the operations carried out in the company are welding and sealing.

The products manufactured by the analyzed company are car seats, produced in four variants. All product variants are similar to one other, except left and right versions and additional elements related to the fixing method in the finished product. The production plans of all variants assume mainly the use of the same machines. Production processes take place in the same production cell and there are differences between single operations. In the analyzed area of production, there are 17 machines in total. In the production of all four variants, the unit times are the same. Data about production process flow is shown in the Fig. 1.

Only the production cell composed of work stations (WST.) A, B, C, D, E and F will be subject to further analysis. The first production stage is assembly (by welding) of stamped metal parts to semi-products and finished products. Stamped sheet metal parts and other components are delivered in special containers to a place allocated at the workstation. At the next stages, the products are lacquered and packed. Each assembly centre is composed of several stations. Stations in an assembly centre are arranged on circumference of a rectangle. There are no buffers (intermediate storage areas) at the workstations. Most often, the technological operations on the workstations are automated welding or sometimes manual welding. Before welding, the operator places components in special fixtures.

Four variants of products are manufactured in series on the assembly line. Sometimes, a change of the production series requires replacement of tooling of the workstations. Each operator can concurrently (at the same time) operate a number of workstations. This concerns mostly the stations located close to each other. Maximum capacity of a work centre depends on the time of the longest lasting technological operation.

To increase production capacity and to simplify the operators' work, cyclical production is required. During cyclical production, for majority of his working hours, the operator executes technological functions in the same sequences. This rule is not valid for the short times at the beginning and at the end of manufacturing a production series.

The presented research work was aimed at developing the algorithm making it possible to determine numbers of operators in individual work centres, guaranteeing their maximum capacities. The algorithm would be especially usable in a company at the time of high absence of employees or in the case when orders from several customers must be executed at the same time.

#### 3. Problem description

We will use the following notation in the problem formulation:

М	– set of machines,
m	– number of machines,
Р	<ul> <li>cell productivity,</li> </ul>
0	- set of operations,
<i>p</i> <sub>i</sub>	– processing time of an operation $i \in O$ ,
n	– number of jobs,
0	– number of operators,
α <sub>k</sub>	- order of operations performing by operator
	k = 1,, 0,
α	- order of operations performing by all operators,
S <sub>j,i</sub>	- starting time of an i-th operation of a job <i>j</i> ,
C <sub>j,i</sub>	– completion time of an i-th operation of a job $j$ ,
С	– cyclic core,
π	– operations order in cyclic core C,
а	– jobs assignment in cyclic core C,
C <sub>k</sub>	<ul> <li>cyclic core of operator k,</li> </ul>
$\pi_k$	– operations order in a cyclic core ${\tt C}$ for the
	operator k,
$\vartheta_k$	– number of operations in the $\pi_k$ ,
a(i)	- assignment of a task to an operation i in the
( )	cyclic core C,
S <sub>i</sub> <sup>(x)</sup>	– starting time of operation i in x-th repetition of
	cyclic core C,
$G(\pi, a)$	- directed graph defined for cyclic core C de-
	scribed by $\pi$ and $a$ ,
V	– set of nodes of the graph $G(\pi, a)$ ,
$E(\pi, a)$	– set of arcs of the graph $G(\pi, a)$ ,
$L_{s^{(x)},t^{(y)}}$	– length of the longest path in $G(\pi, a)$ from the
	node s in x-th cyclic core to the node t in y-th
	cyclic core,
	node t in y-th cyclic core,
$L^*(\pi, a)$	<ul> <li>lower bound of the cycle time for the cyclic core</li> </ul>
	С,
$T(\pi, a)$	<ul> <li>minimal cycle time for the cyclic core C.</li> </ul>

Each production centre in the assembly line consists of *m* workstations from the set  $M = \{1, 2, ..., m\}$ . In the centre, *n* identical products of the series are manufactured in the cyclical way. Each product is processed in the workstations in the sequence 1, 2, ..., m. Thus, all products generate the production order consisting of *m* technological operations. No operational buffers are present between the workstations [17,18] (so called no-store constraint). Execution time of the production order on the station  $i \in M$  is  $p_i > 0$  (minutes) and maximum capacity of the centre is [7]:

$$P = 60/\max_{i \in M} p_i \tag{1}$$

pieces per hour.

All the employees have qualifications enabling them to work on any workstation. On the workstation, the employee performs technological operations or supervises them, all the time staying at the workplace. At a given moment, only one product can be processed on the station and the employee can operate only one station. Technological operations performed on the machines cannot be discontinued. Each technological operation executed by the operator in the system will be designated by the pair (j,i), where j =1, 2, ..., n means the number of a subsequent product and  $i \in M$ means the workstation. Let  $\alpha_k$  mean a sequence of executing technological operations by the operator k = 1, 2, ..., o. The sequence of all the operations executed by the operators in a production centre is described by the expression  $\alpha = (\alpha_1, ..., \alpha_o),$   $\alpha_k = (\alpha_k(1), ..., \alpha_k(n_k)),$  where  $\alpha_k(s) = (j_k(s), i_k(s)), s = 1, ..., n_k$  and  $n_k$  means number of the operations executed by the operator k. Let  $S_{j,i}$  and  $C_{j,i}$  denote the beginning (completion) time of execution of the task j in the workstation i. The time schedule of executing the tasks for a given sequence  $\alpha$  must fulfil the following inequalities [7]:

$$S_{j,i} \ge 0, \ j = 1, 2, \dots, n, \ i = 1, 2, \dots, m,$$
 (2)

$$S_{j,i} \ge C_{j,i-1}, \ j = 1, 2, ..., n, \ i = 2, 3, ..., m,$$
 (3)

$$C_{j,i} = S_{j,i} + p_{j,i}, j = 1, 2, ..., n, i = 1, 2, ..., m,$$
 (4)

$$S_{j_k(s),i_k(s)} \ge C_{j_k(s-1),i_k(s-1)}, \ s = 2, 3, \dots, n_k, \ k = 1, 2, \dots, o, \tag{5}$$

$$\begin{split} & S_{j_k(s),i_k(s)} \geq S_{j_k(s-1),i_k(s-1)+1}, \, i_k(s-1) < m, \, s=2,3,\ldots,n_k, \\ & k=1,2,\ldots,o. \end{split}$$

The inequality (2) anchors the time schedule at the moment 0. The inequality (3) means that the operation of the given task cannot be started before the previous operation of this task is completed. The inequality (4) means that the operation cannot be discontinued. It results from the inequality (5) that the s-th operation executed by the operator k can be started only after the previous operation is completed. The "no storage" condition is modelled by the inequality (6) and means that execution of the operation can be started only after the workstation is released by the previous task, i.e. at the moment when execution of the next operation of this task is started.

The sequence of operations  $\alpha$  to be executed by the operators is only permissible, when the system of inequalities is solvable (2–6). Permissibility of  $\alpha$  can be verified and time schedule of executing the operations in the sequence  $\alpha$  can be determined at the moment O(n,m) by building, on the grounds of the inequalities (2–6), a directed graph (like it was done in Refs. [7,11],) described in detail in Section 3.2.

The sequence  $\alpha$  will be called the cyclic sequence if it is constructed on the basis of a cyclic core. The cyclic core consists of cyclically performed *m* operations from one or more tasks. Let  $O = \{1, ..., m\}$  be the set of operations belonging to a cyclic core. The order of executing operations from the cyclic core will be denoted by the symbol

$$C = (C_1, \dots, C_o), \tag{7}$$

 $C_k = (C_k(1), \ldots, C_k(\vartheta_k)), \ m = \sum_{k=0}^{o} \vartheta_k$  where  $\vartheta_k$  is the number of operations performed by the operatork. Then the order  $\alpha$  is a core assembly, i.e.

$$\alpha = \alpha^0 CC \dots C\alpha^*, \tag{8}$$

where  $\alpha^0$  — stands for the preliminary order, CC ... C — repeatedly repeating the cyclic core,  $\alpha^*$  — the ending order.

The most widely recognized in the literature is fully automated production in cells, in which the transport of products is carried out by an industrial robot [9,10,13,15].

#### 3.1. An example

The considered production cell consists of m = 6 machines and is operated by o = 4 operators. The names of technological activities and machining times (in seconds) are given in Table 1. Operators 1, 2, 3 perform only one operation respectively 3, 4 and 6, while operator 4 operations 1, 2 and 5. The system performs n = 7 products [7].

Fig. 2 illustrates the schedule for performing operations in the form of Gantt chart. The execution schedule of the technological operations resulting from the sequence  $\alpha^0$  and  $\alpha^*$  is marked with the dotted line and the schedule resulting from the first and the last occurrence of the cyclical core is marked with the solid line. The dashed line means a delay resulting from a blockage of the workstation caused by absence of a storage buffer. In addition, the beginning and completion moments of all the operations are gathered in Table 2.

Table 1 – List of technological operations in the analysed production centre.					
Order of operations performed by an operator	Order of operations performed by an operator	Order of operations performed by an operator			
1	Sealing	45			
2	Sealing	26			
3	Welding	134			
4	Manual welding	104			
5	Welding	34			
6	Welding	104			



Fig. 2 – The schedule of operations performed by operators.

It should be noted that the operator No. 4 operates 3 workstations: 1, 2 and 5. So, these stations should be located close to each other. For example, the stations A and B could be located side by side and the station E could be opposite to the other ones. Time schedule of that operator (4) is the most complicated, especially at the beginning of the work. First of all, the operator executes the first two operations of the first task, then the first two operations of the second task, then he must wait until the second station is vacant (i.e. till the moment when the task 2 on the station A is started) and execute the first two operations of the operator 4 are performed cyclically and are composed of three operations on the stations E, A and B and the break related to waiting until operations of the operator 2 are completed.

#### 3.2. Determination of the cycle time

Cycle time can be determined on the basis of *r* successively repeated cyclic cores C. Let us denote by  $\pi = (\pi_1, \dots, \pi_o), \pi_k = (\pi_k(1), \dots, \pi_k(n_k))$  executing order of operations performed by the operator *k* on the basis of the core C. Additionally, by the symbol  $S_i^{(x)}$  let us represent the time of execution beginning of

Table 2 – Production schedule executed by the operators.							
Par	t	Operations					
		1	2	3	4	5	6
1	S <sub>1i</sub>	0	45	71	205	309	343
	$C_{1i}$	45	71	205	309	343	447
2	S <sub>2i</sub>	71	116	205	339	443	477
	C <sub>2i</sub>	116	142	339	443	477	581
3	S <sub>3i</sub>	205	250	339	473	577	611
	C <sub>3i</sub>	250	276	473	577	611	715
4	S <sub>4i</sub>	343	388	473	607	711	745
	$C_{4i}$	388	414	607	711	745	849
5	$S_{5i}$	477	522	607	741	845	879
	C <sub>5i</sub>	522	548	741	854	879	983
6	S <sub>6i</sub>	611	656	741	875	979	1013
	C <sub>6i</sub>	656	682	875	979	1013	1117
7	S <sub>7i</sub>	745	790	875	1009	1113	1147
	C <sub>7i</sub>	790	816	1009	1113	1147	1251

the operation i in the x-th cycle. Constraints (1)–(6) can be transformed into the following inequality:

$$S_{i}^{(x)} \ge S_{i-1}^{(x-a(i)+1)} + p_{i-1}, i = 2, 3, ..., m, x = 1, 2, ..., r,$$
 (9)

where a(i) denotes a number of task assigned to an operation i in the core C.

$$S_{\pi_k(s)}^{(x)} \ge S_{\pi_k(s-1)}^{(x)} + p_{\pi_k(s-1)}, s = 2, 3, \dots, n_k, k = 1, 2, \dots, o, x = 1, 2, \dots, r,$$
(10)

$$\begin{split} S^{(x)}_{\pi_k(s)} &\geq S^{(x)}_{\pi_k(s-1)+1}, \ \pi_k(n_k) + 1 \leq m; s = 2, 3, \dots, n_k, \ k = 1, 2, \dots, o, \\ x = 1, 2, \dots, r, \end{split}$$

$$S_{\pi_k(1)}^{(x)} \ge S_{\pi_k(n_k)}^{(x-1)} + p_{\pi_k(n_k)}, s = 2, 3, \dots, n_k, k = 1, 2, \dots, o, x = 2, 3, \dots, r,$$
(12)

$$\begin{split} S^{(x)}_{\pi_k(1)} &\geq S^{(x-1)}_{\pi_k(n_k)+1}, \ \pi_k(n_k) + 1 \leq m; \ s = 2, 3, \dots, n_k, \ k = 1, 2, \dots, o, \\ x = 2, 3, \dots, r. \end{split}$$

It is easy to see that inequality (9) is a combination of (3) and (4), inequality (10) is a combination of (5) and (4), while inequality (11) results from (4) and (2). All these inequalities describe time relations in one core. Inequalities (12) and (13) are equivalent to inequalities (10) and (11) and combine the two following cores.

For a pair  $(\pi, a)$  which follows from the core *C* we construct an elementary directed graph  $G(\pi, a) = (V, E(\pi, a))$  with the set of vertices *V* and the set of arcs  $E(\pi, a)$ . The vertices set *V* we define as follows:

$$V = \bigcup_{x=1}^{r} \left\{ i^{(x)} : i \in O \right\}.$$
 (14)

A vertex  $i^{(x)} \in V$  represents an operation  $i \in O$  executed in the x -th core. A vertex  $i \in O$  is weighted with the value  $p_i$ , i.e. the duration time of an operation i.

A set of arcs we define as follows:

 $\mathsf{E}(\pi, a) = \mathsf{R}(a) \cup \mathsf{E}(\pi) \cup \mathsf{F}(\pi) \cup \mathsf{E}^*(\pi) \cup \mathsf{F}^*(\pi).$ 

The set E includes arcs of five subsets, defined as follows:

$$R(a) = \left\{ \left( i^{(x-a(i)+1)}, i^{(x)} \right) : i = 2, 3, \dots, m, \ x = 1, 2, \dots, r \right\}.$$
 (15)

Arcs from the set R(a) are weighted with the value zero and they are modeling technological constraints which follows from the inequality (10):

$$\begin{split} E(\pi) &= \{(\pi_k(s-1)^{(x)},\pi_k\left(s^{(x)}\right):s=2,3,\ldots,n_k,\\ k=1,2,\ldots,o,\,x=1,2,\ldots,r\}. \end{split}$$

Arcs from the set  $E(\pi)$  represent an order of operations execution by operators, they are weighted with the value zero and they are connected with the constraint (12). Arcs of the set

$$\begin{split} F(\pi) &= \{ ((\pi_k(s-1)+1)^{(x)}, \pi_k\left(s^{(x)}\right) : s=2,3,\ldots,n_k, \\ k=1,2,\ldots,o, \, x=1,2,\ldots,r \} \end{split}$$

are modelling no-store (blocking) constraint. An arc  $\{((\pi_k(s-1)+1)^{(x)}, \pi_k(s)^{(x)} \in F(\pi) \text{ has the weight } -p_{\pi_k(s-1)+1} \text{ and is connected with the constraint (12).}$ 

Other arcs sets connect consecutive cores of cyclic order:

$$\mathbf{E}^{*}(\pi) = \{(\pi_{k}(n_{k})^{(\mathbf{x})}, \pi_{k}(\mathbf{1})^{(\mathbf{x})}: \ k = 1, 2, \dots, o, \ \mathbf{x} = 1, 2, \dots, r\}.$$
(18)

Arcs from the set  $E^*(\pi)$  are related the arcs from the set  $E(\pi)$ , while arcs

$$F^{*}(\pi) = \{ ((\pi_{k}(n_{k}) + 1)^{(x)}, \pi_{k}(1)^{(x)} : k = 1, 2, \dots, o, x$$
  
= 1, 2, \dots, r \} (19)

corresponds to arcs from the set  $F(\pi)$ . The arc  $\{((\pi_k(n_k) + 1)^{(x)}, \pi_k(1)^{(x)} \in F^*(\pi) \text{ has the (negative) weight } - p_{\pi_k(n_k)+1}.$ 

The Fig. 3 shows the graph  $G(\pi, a)$  for 7 production cycles from the example presented in Section 3.1. There are two

numbers in each node: job number and operation number. In Fig. 3 there are three types of arcs: diagonal — modeling of technological limitations, horizontal — modeling of the sequence of operations performed by operators, and interrupted — modeling of binding 'no-store'.

We denote the length of the longest path from the vertex  $s^{(x)}$  to the vertex  $t^{(y)}$  by  $L_{s^{(x)},t^{(y)}}$ . Using the  $G(\pi, a)$  graph, we can formulate some useful properties.

**Theorem 1.** Let  $G(\pi, a)$  be a graph constructed for the core *C*. If the cyclic core *C* determines a feasible order of operations execution by operators, then the graph  $G(\pi, a)$  does not includes cycles of the positive length.

**Proof.** (a contrary) Let us assume, that  $G(\pi, a)$  has a cycle of a positive length. It means, that for a pair of vertices  $(s^{(x)}, t^{(y)})$  there is no longest path of a finite length. Thus, it is impossible to indicate finite terms of starting of both operations  $s^{(x)}$  and  $t^{(y)}$  which is contrary to the assumption that *C* is a feasible core.

**Proposal 1.** Let the graph  $G(\pi, a)$  be constructed for a core C and let it does not include cycles of the positive length. Then the cyclic core C denotes a feasible order of operations execution by operators.

Theorem 2. If for a pair of vertices  $s^{(x)},t^{(y)}\in V$  we have  $L_{s^{(x)},t^{(y)}}<\infty$  ,  $L_{s^{(x)},t^{(y)}}<\infty$  , then

$$S_t^{(y)} \ge S_s^{(x)} + L_{s^{(x)}, t^{(y)}}.$$
(20)

**Proof.** The proof results directly from the construction of the  $G(\pi, a)$  graph and constraints (l2.1)-(l2.5).

Let

$$L^{*}(\pi, a) = \max_{2 \le y \le r} \max_{1 \le k \le 0} \left\{ \frac{L_{\pi_{k}(1)^{(1)}, \pi_{k}(1)^{(y)}}}{y - 1} : L_{\pi_{k}(1)^{(1)}, \pi_{k}(1)^{(y)}} < \infty \right\}$$
(21)

where  $r = o + \max_{1 \le i \le m} a(i)$ . Then the following theorem occurs.



Fig. 3 – Graph  $G(\pi, a)$  designed for the cyclic core C.

7

**Theorem 3.** For the order of operations execution  $\pi$  and the assignmet of operations to tasks *a*, the cycle time

$$T\pi, a = L^*\pi, a. \tag{22}$$

Equality (22) can be proved similarly, as Property 2 from the work [16].

#### 4. Optimization algorithm

Due to the relatively small size of the data instance, we have chosen the Branch and Bound method as the basis of the optimization algorithm, which gives us an optimal solution to the problem under consideration. The method is well known, and its details can be found e.g. in Ronconi's article [17]. Due to the relatively small number of machines in the socket, even a complete inspection was possible. However, we decided to use the Branch and Bound method due to the additional time gain. Actually, it was not necessarily necessary — the number of machines in the cell does not exceed a dozen or so in practice — and the possibilities of "boundary" B & B methods are currently looming around 40–50 decision variables, which is a value met with a large margin.

At the very beginning it should be noted that after a sufficient number of cyclic core repetitions, the initial moments for the respective operations belonging to two successive cores x-1 and x are separated by the same period of time T, i.e. [7]:

$$S_{x,i} = S_{x-1,i} + T, i = 2, 3, ..., m.$$
 (23)

Hereinafter, the period T will be named the cycle time. It can be easily seen that the production centre works with full capacity (its capacity is limited by the time of the longest executed operation), if the cycle time is equal to  $\max_{k \in M} p_k$ .

Length of the cycle time for a cyclical core  $\pi$  be calculated by generation of the sequence  $\alpha$  and determination of the value T in the steady state. Unfortunately, such a method requires generation of  $\alpha$  on the grounds of a large number of core repetitions. The cycle time can be determined much faster using a method similar to that used in Refs. [2,3,5,8,16,19].

In a natural way, the lower limit of the number of operators is FT/T, where  $FT = \sum_{s=1}^{m} p_s$  is the time the task flows through the production system, while the upper limit is equal to the

number of machines. The minimum number of operators can be determined by searching half of the minimum number of operators for which a schedule with cycle time  $T = \max_{k \in M} p_k$ can be constructed [7]. During the operation of the algorithm, cyclic cores fulfilling the condition [7]

$$\max_{k \in M} \sum_{i=1}^{\vartheta_k} p_{\pi_k(i)} > \mathsf{T},$$
(24)

were not generated.

Of course, among the cores that meet this condition may be those for which it is impossible to construct a cyclical schedule with the cycle time *T*. The algorithm ends when the core is found with the cycle time *T* or after generating all cyclic cores for the given number of operators.

### 5. Computational experiments

Computerised analysis was performed on 20 exemplary products for that the following data were acquired [7]:

- 1 m number of technological operations,
- 2 FT =  $\sum_{s} p_{s}$  flow time of a task through the production system;
- 3  $T = \max_{k \in M} p_k$  cycle time.

Next, minimum number of employees required for execution of the task N0 was determined, as well as the following quantities [7]:

FT/T — lower-bound estimate of the number of employees,
 m/NO — ratio of the number of work stations to that of employees.

The optimizing algorithm was implemented in the C++ language in the environment Visual Studio 2010. It was run on a computer with the processor Intel Core i7 2.4 GHz, under the operating system Windows 8.1. For each product, running time of the optimizing algorithm did not exceed several dozen seconds [7].

It arises from the experimental results collected in Table 2 that, in all the 20 cases, the algorithm determined the optimum number of operators equal to the lower-bound estimate. It should be noted that it is trivial to determine a lower-bound estimate of the number of employees, but assignment of operators to workstations and technological operations, as well as determination of a time schedule for most products is very difficult and requires using the suggested algorithm [7].

Values of the coefficient m/NO range between 1 and 2.6. The value means that, for some products, number of the employees is equal to that of the workstations, so that each employee operates one workstation only. For the highest value of the ratio, an operator operates on average as many as 2.6 workstations, i.e. 3 operators operate, with full capacity, 8 workstations in a production centre.

#### 6. Summary

In the paper, the problem of determining the smallest number of operators necessary to execute the production schedule in a multi-station production centre is analysed. To that end, the original concept of a so-called cyclical core simplifying the operation sequence of a number of workstations by the production employees is introduced. A method of determining the sequence of all the technological operations and their execution schedule is suggested.

An optimizing algorithm based on the method of complete overview of all the cyclical cores is suggested. The algorithm eliminates generation of cyclical cores for that the production

Table 3 – Results of experimental research [7].						
Product	m	FT	Т	NO	FT/T	m/NO
1	9	10.197	2.390	5	5	1.8
2	10	11.049	2.390	5	5	2.0
3	12	12.579	2.989	5	5	2.4
4	12	12.579	2.989	5	5	2.4
5	12	13.431	2.989	5	5	2.4
6	5	19.129	10.600	2	2	2.5
7	5	19.563	10.600	2	2	2.5
8	5	13.258	9.204	2	2	2.5
9	6	7.452	2.232	4	4	1.5
10	7	13.319	4.061	4	4	1.7
11	8	14.399	4.061	4	4	2.0
12	8	17.498	7.416	3	3	2.6
13	9	18.212	7.416	3	3	3.0
14	6	13.002	4.410	3	3	2.0
15	5	18.09	12.766	2	2	2.5
16	3	5.202	2.487	3	3	1.0
17	3	4.044	1.825	3	3	1.0
18	3	4.044	1.825	3	3	1.0
19	3	4.405	2.186	3	3	1.0
20	3	12.321	5.187	3	3	1.0

schedule would not guarantee full-capacity work of the work centre. Its suitability was verified by analysis of 20 products manufactured in a company producing seats for passenger cars. For all the products, a minimum number of operators equal to the lower-bound estimate and their acceptable work schedule were determined (Table 3).

Operational planning in cyclic production systems is critical for the production efficiency of large companies. In today's widely applied in information systems of ERP class used to support management, particularly important are numerically efficient methods and algorithms for solving the new optimization problems derived from real manufacturing systems which constitute "intelligent engines" for these support systems. In majority of cases, practical problems generate NP-hard combinatorial optimization problems (multi-extremal), which due to the complexity (e.g. size, criteria, time constraints), can be modeled and solved with the use of the latest methods of discrete optimization developed in recent years (and still undergoing the process of constant development), in particular, effective multithread, parallel, scattered and bio-inspired methods.

## **Conflicts of interest**

Non-declared.

## Acknowledgement

The paper was partially supported by the National Science Centre, Poland, grant OPUS no. DEC 2017/25/B/ST7/02181.

#### REFERENCES

- [1] A. Ambroziak, M. Korzeniowski, Using resistance spot welding for joining aluminium elements in automotive industry, Arch. Civ. Mech. Eng. 10 (1) (2010) 5–13.
- [2] G. Bocewicz, R. Wójcik, Z. Banaszak, Cyclic steady state refinement, Springer, International Symposium on Distributed Computing and Artificial Intelligence (2011) 191–198.
- [3] W. Bozejko, A. Gnatowski, J. Pempera, M. Wodecki, Parallel tabu search for the cyclic job shop scheduling problem, Comput. Ind. Eng. 113 (2017) 512–524.
- [4] W. Bozejko, Z.M. Hejducki, M. Wodecki, Applying metaheuristic strategies in construction projects management, J. Civ. Eng. Manag. 8 (5) (2012) 621–630.
- [5] W. Bozejko, Z. Hejducki, M. Wodecki, Flowshop scheduling of construction processes with uncertain parameters, Arch. Civ. Mech. Eng. 19 (1) (2019) 194–204.
- [6] W. Bozejko, M. Wodecki, Parallel genetic algorithm for minimizing total weighted completion time, Lect. Notes Comput. Sci. 3070 (2004) 400–405.
- [7] W. Bozejko, J. Pempera, M. Wodecki, Minimization of the number of employees in manufacturing cells, in: M. Graña, et al. (Eds.), International Joint Conference SOCO'18-CISIS'18-ICEUTE'18, Advances in Intelligent Systems and Computing, 771, Springer, Cham, 2018.
- [8] A. Burduk, W. Bozejko, J. Pempera, K. Musial, On the simulated annealing adaptation for tasks transportation optimization, Log. J. IGPL 26 (6) (2018) 581–592.
- [9] A. Che, C. Chu, Multi-degree cyclic scheduling of a no-wait robotic cell with multiple robots, Eur. J. Oper. Res. 199 (1) (2009) 77–88.
- [10] I.G. Drobouchevitch, S.P. Sethi, C. Sriskandarajah, Scheduling dual gripper robotic cell: one-unit cycles, Eur. J. Oper. Res. 171 (2) (2006) 598–631.
- [11] J. Grabowski, J. Pempera, The permutation flow shop problem with blocking. A tabu search approach, Omega. 35 (2007) 302–311.
- [12] B. Karwat, Optimization of production schedules in the steel production system, Arch. Civ. Mech. Eng. 12 (2) (2012) 240–252.
- [13] E. Levner, V. Kats, V.E. Levit, An improved algorithm for cyclic flowshop scheduling in a robotic cell, Eur. J. Oper. Res. 97 (3) (1997) 500–508.
- [14] A. Munier Kordon, A graph-based analysis of the cyclic scheduling problem with time constraints: schedulability and periodicity of the earliest schedule, J. Sched. 14 (1) (2011) 103–117.
- [15] K. Musial, D. Górnicka, A. Burduk, Improvement of production process scheduling with the use of heuristic methods., in: A. Burduk, E. Chlebus, T. Nowakowski, A. Tubis (Eds.), Intelligent Systems in Production Engineering and Maintenance, Advances in Intelligent Systems and Computing, 835, Springer, Cham, 2018.
- [16] J. Pempera, C. Smutnicki, Open shop cyclic scheduling, Eur. J. Oper. Res. 269 (2018) 773–781.
- [17] D.P. Ronconi, A branch-and-bound algorithm to minimize the makespan in a flowshop with blocking, Ann. Oper. Res. 138 (2005) 53–65.
- [18] D.P. Ronconi, A note on constructive heuristics for the flowshop problem with blocking, Int. J. Prod. Econ. 87 (2004) 39–48.
- [19] B. Skolud, D. Krenczyk, M. Zemczak, Multi-assortment rhythmic production planning and control, IOP Conf, Ser. Mater. Sci. Eng 95 (1) (2015) 1–6, 012133..