10.24425/acs.2020.133498

Archives of Control Sciences
Volume 30(LXVI), 2020
No. 2, *pages 213–231*

Method of rescheduling for hybrid production lines with intermediate buffers

MAREK MAGIERA

A method of creating production schedules regarding production lines with parallel machines is presented. The production line setup provides for intermediate buffers located between individual stages. The method mostly concerns situations when part of the production machines is unavailable for performance of operations and it becomes necessary to modify the original schedule, the consequence of which is the need to build a new schedule. The cost criterion was taken into account, as the schedule is created with the lowest possible costs regarding untimely completion of products (e.g. fines for delayed product completion). The proposed method is relaxing heuristics, thanks to which scheduling is performed in a relatively short time. This was confirmed by the presented results of computational experiments. These experiments were carried out for the rescheduling of machine parts production.

Key words: rescheduling, scheduling, heuristic, flowshop, hybrid production line, buffer management

1. Introduction

The presented method concerns multi-stage production lines. Every stage is a collection of machines which work in parallel. Between these stages, intermediate buffers are located, where the products can wait for performance of consecutive operations. On these production lines, operations regarding various types of products can be performed simultaneously.

The purpose of the paper is to present a method intended for reconstruction of production schedules in the case of impossibility to perform production orders according to a pre-determined schedule. This makes the method useful in the case of unexpected unavailability of certain production machines, if a new production schedule has to be compiled in a relatively short time. The short compilation

The Author (E-mail: mmagiera@zarz.agh.edu.pl) is with AGH University of Science and Technology, Department of Operations Research, al. Mickiewicza 30, 30-059 Kraków, Poland.

Received 21.09.2019. Revised 25.05.2020.

time of the new schedule was achieved by using a linear relaxation of the integer programming task.

The subject matter of the paper concerns scheduling, as each rescheduling is a distribution of tasks in space (between machines) and in time. Frequently, rescheduling must be done in a very short time. Then, this activity is called online scheduling (Suwa and Sandoh, 2013).

Katragjini et al. (2013) distinguished two groups of disturbances to the original schedule, resulting in the need for rescheduling:

Capacity disruptions, i.e. disturbances related to manufacturing resources such as production machine breakdowns, unavailability of tools, machine operator absence, etc.

Order disruptions, i.e. change in production priority, job-related disturbances such as rush jobs, job cancellation, raw material shortage, etc.

The literature concerning distribution of operation between machines and determining starting times of these operations is extensive. Soualhia et al. (2017) described an overview of IT tools used for task scheduling. An overview of the task scheduling in hybrid systems, which this article pertains to, was made by Ruiz and Vasquez-Rodriguez (2010). They analysed over two hundred papers. Their analysis of the applied optimality criteria showed that the time criteria predominate, constituting over 90% of the applied criteria. Among these criteria, schedule length, also referred to as operation scheduling length, dominates. The second group of criteria included in scheduling are the cost criteria. The criterion applied in the developed method concerns this very group of criteria. The proposed method is intended to minimize the costs related to untimely performance of operations concerning the products. These costs include fines for delayed product completion, as well as costs of storing products made before the date specified in the order.

Of course, there are significant connections between the optimality criteria, such as, for instance, the lower the untimeliness, the lower the costs, including fines for delayed completion of the products and the costs of storing products completed prematurely. Selection of the optimality criteria results from the specificity of the model being solved. Rescheduling according to the developed method contributes to designation of such schedules where the product completion dates are as close as possible to the dates specified in the original schedule.

The scheduling also employs multi-criterial methods. One example thereof is the two-criterial method, taking into account the time and cost criteria, created by Behnamin and Fatemi Ghomi (2011).

Production line scheduling makes use of accurate and approximate methods. The developed method is an approximate one. The schedules are determined in a relatively short time, yet at the expense of a certain deviation from the optimal solution. The rudiments of creating approximate algorithms and the issues regarding construction of approximate algorithms were described, e.g., by

Gonzales (2007). Disadvantages of the approximate methods include: premature convergence to the relative extremum, as well as stagnation in the search for solution sets. These disadvantages contributed to the development of hybrid methods. The idea of hybrid methods was described in detail by Benhamin and Fatemi Ghomi (2011), whereas an overview of the hybrid methods was presented by Ribas et al. (2010). The problem of applying hybrid methods to scheduling was described by Seck-Tuoh-Mora et al. (2019).

Due to the concept of the solving the problem, production line rescheduling methods are divided into monolithic and hierarchical. The monolithic methods are characterized by a global approach to the problem. In the alternate – hierarchical – conception, also referred to as multi-level, the problem in question is divided into partial tasks, which are solved successively. The main reason behind using this conception is the possibility of solving sizeable problems in a relatively short time. This is done at the expense of a certain deviation from the optimum, which means these methods are classified as approximate. The advantages and disadvantages of both concepts are described, for instance, in a publication edited by Castillo et al. (2005). The results of computational experiments concerning comparison of these concepts in the case of production line scheduling were collated by Magiera (2015).

The method presented in the paper is monolithic in nature. The applied relaxing technique contributed to determination of approximate solutions in very short calculation time. Relaxation concerns the mathematical model intended for assembly scheduling. The developed method employs linear programming. This sort of mathematical programming is more and more often used in production planning at the operating level, the result of which is production scheduling (Leung, 2004). According to the classification of mathematical problems (Chan et al., 2017), the method presented herein concerns production planning, as part of which task scheduling is performed.

Creation of mathematical models concerning the developed methods was inspired, e.g., by the works of Pindeo (2009), Ronconi & Birgin (2004, 2012), as well Tóth et al. (2018). These works show very good perspectives for using linear programming in production planning. Rapid development of computer technology and software favours this.

The presented mathematical model and heuristics concern any number of machines set up in a production line. Currently, many algorithms are dedicated to specific configurations of the machinery stock, e.g. they concern a strictly specified number of machines. E.g. Furmańczyk and Kubale (2017) developed a scheduling method intended for four machines, in which they effectively used the graph theory.

The schedules created using the presented method concern Flexible Manufacturing Systems (FMS). In these systems, operations regarding various types of products can be performed simultaneously. The issues concerning Flexible Manufacturing Systems were presented, e.g., by Yadov & Jajswal (2017).

Unlike scheduling, the literature concerning rescheduling is not particularly abundant. The papers are mostly concerned with rescheduling of services in specific fields. They concern, e.g., creation of new schedules regarding transport, distribution of employees between customer service stations, surgeries performed on patients in hospitals. Veiera et al. (2003) emphasized the need for rescheduling in the case of production systems. Their paper discusses studies that show how rescheduling affects the performance of a manufacturing system, and it concludes with a discussion of how understanding rescheduling can bring closer some aspects of scheduling theory and practice.

This paper falls into the described current of studies regarding rescheduling. The presented rescheduling method, dedicated to production systems with hybrid production lines, is mostly to be used to create new schedules in the case of any disturbance to the production plan. The economic aspect should also be emphasized. These schedules are created with a view to maximally reduce untimeliness of task performance, which has a positive impact on the costs incurred. Timely performance of tasks means no fines for delays and lower storage costs. Taking these economic aspects into account is significant in the case of any market characterized by considerable competitiveness, where the timeliness of completing production tasks is of significant importance. The aforesaid aspects contributed to development of a rescheduling method, described in detail further on.

The detailed presentation of the developed method is included in the following sections of the paper. Section 2 is dedicated to the description of the problem and the concept of its solution. Section 3 features linear mathematical models. Relaxation heuristic is described in Section 4. The results of computational experiments used for verification of the developed mathematical model and the heuristic algorithm are presented in Section 5.

2. The description of the problem and linear mathematical model

The rescheduling task concerns a unidirectional, multi-stage production line. A model setup of such a production line is shown in Fig. 1. Intermediate buffers are located between the stages. Each products encumbers no more than one machine of the given stage. Certain stages may be omitted.

Figure 2 shows the block diagram of the monolithic rescheduling method. The series of tasks includes parameters describing the machinery stock, products as well as costs associated with untimely completion of the products. The sum of costs incurred on account of this untimeliness is minimized.

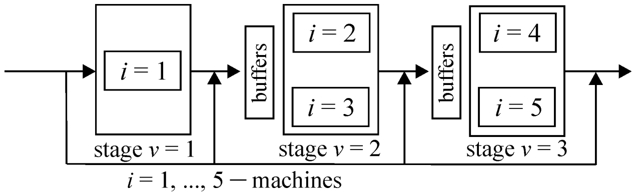

Figure 1: Diagram of hybrid production system with intermediate buffers

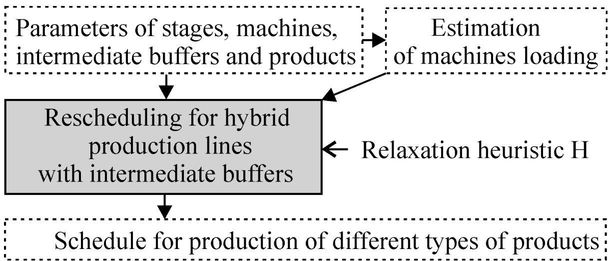

Figure 2: Block diagram of the monolithic rescheduling method

Table 1 presents parameters, sets and variables used in the mathematical model applied in the developed method. It should be noted that the product set K was divided into two disjoint subsets. The K^1 subset includes products for which the operations are to be carried out as per the original schedule. Whereas subset K^2 contains products which are to be made in accordance with the new schedule.

One of distinctive features of the developed method is that the ranking lengths are divided into periods (unitary intervals), the number of which is H (Table 1). The number of the considered periods should be greater than the estimated ranking length, it can be determined, e.g., using the procedure described in Magiera's paper (2013).

The designations of parameters and variables shown in Table 1 were used in the following mathematical model. This model was utilized in the developed method.

The mathematical model M:

$$\text{Minimize:} \quad \sum_{k\in K^2} \left(c_k^1 e_k^1 + c_k^2 e_k^2 + c_k^3 w_k\right) \tag{1}$$

$$\text{subject to:} \quad \sum_{i\in I} \sum_{l\in L:\, n_{il}=1} q_{ijkl} = p_{jk}\ ; \qquad (k,j) \in O, \tag{2}$$

$$q_{ijkl} \geqslant 1\ ; \quad i \in I;\ \ j \in J;\ \ k \in K^1;\ \ l \in L\colon (i,j,k,l) \in Z, \tag{3}$$

$$\sum_{l\in L} \sum_{k\in K} q_{ijkl} = 0; \qquad (i,v) \in F;\ \ j \in J;\ \ (v,j) \notin V_j\,, \tag{4}$$

Table 1: Summary of indices, parameters and variables used in the method

Indices:		
i	–	machine; $i \in I$;
j	–	operation; $j \in J$;
k	–	type of product; $k \in K$;
l	–	period; $l \in L = \{1, \ldots H\}$;
v	–	stage; $v \in V$;
Parameters and sets:		
c_k^1	–	cost incurred in a unit of time, resulting from delayed product completion $k \in K^2$;
c_k^2	–	cost incurred in a unit of time, resulting from premature product completion $k \in K^2$;
c_k^3	–	cost equal to the fine for failure to meet the latest, permissible deadline for product completion $k \in K^2$;
d_v	–	capacity of the buffer located before stage v;
$g_{\varepsilon v}$	–	transport time between machines at stage ε and at stage v;
m_v	–	number of machines at stage v;
n_{il}	=	1, if machine i is available during period l, otherwise $n_{il} = 0$;
p_{jk}	–	processing time for operation $j \in J$ for type of product k;
t_k^1	–	ordered completion time for product $k \in K^2$, subject to rescheduling;
t_k^2	–	the latest deadline for completing the product $k \in K^2$, after which a unit fine is charged;
ρ_k	–	system readiness for performing operation concerning the product $k \in K^2$;
K^1	–	the set of products which are to be made in original with the new schedule;
K^2	–	the set of products which are to be made in accordance with the new schedule;
F	–	the set of pairs (i, v), in which machine i is located in the stage v;
V_j	–	the set of the stages in which the machine are capable of execution of operation j;
O	–	the set of pairs (k, j), in which the operation $j \in J$ is required for type of product $k \in K^2$;
P	–	the set of pairs (k, j), in which the operation $j \in J$ is the last operation of product $k \in K^2$;
R	–	the set of elements (k, r, j), in which operation j is executed immediately before task r; $r, j \in J$ and operations are required for type of product $k \in K$;
Z	–	the set of elements (i, j, k, l), in which in accordance with the original schedule product $k \in K^1$ is assigned to machine i to perform operation j in period l;
Variables:		
e_k^1	–	time of delay in completing the product $k \in K^2$;
e_k^2	–	time concerning premature completion of the product $k \in K^2$; $e_k^2 = t_k^1 - z_k$;
q_{ijkl}	=	1, if product k is assigned to machine i to perform operation j in period l, otherwise $q_{ijkl} = 0$;
w_k	=	1, if the latest deadline for completing the product $k \in K^2$ was exceeded, otherwise $w_k = 0$;
y_{vkl}	=	1 if product k is assigned to buffer positioned before stage v, during period l, otherwise $y_{vkl} = 0$;
z_k	–	time of completing production of the product $k \in K^2$.

$$\sum_{i\in I}\ \sum_{j\in J:\ (k,j)\in O} q_{ijkl} \leqslant 1; \qquad k\in K^2;\ \ l\in L, \tag{5}$$

$$\sum_{j\in J}\sum_{k\in K} q_{ijkl} \leqslant n_{il}\,; \qquad i\in I;\ \ l\in L, \tag{6}$$

$$lq_{ijkl} - fq_{ijkf} \leqslant p_{jk} - 1 + (H+1)\left(1 - q_{ijkf}\right); \\ i\in I;\ \ (k,j)\in O;\ \ f,l\in L;\ \ f<l, \tag{7}$$

$$q_{ijkl} + q_{\tau rkf} \leqslant 1; \\ (i,v),\ (\tau,v)\in F;\ \ \tau\neq i;\ \ r,j\in J;\ \ r\neq j;\ \ k\in K^2;\ \ l,f\in L, \tag{8}$$

$$q_{ijkl} + q_{\tau jkf} \leqslant 1; \quad i,\tau\in I;\ \ \tau\neq i;\ \ (k,j)\in O;\ \ f,l\in L, \tag{9}$$

$$\sum_{i\in I}\sum_{l\in L}\frac{q_{ijkl}}{p_{jk}} - \sum_{i\in I}\sum_{l\in L}\frac{q_{irkf}}{p_{rk}} - \frac{p_{jk}+p_{rk}}{2} \geqslant 0; \qquad (k,r,j)\in R, \tag{10}$$

$$lq_{ijkl} - fq_{\tau rkf} - 1 \geqslant g_{\varepsilon v} - (H+1)\left(1-q_{ijkl}\right); \qquad (\tau,\varepsilon),\ (i,v)\in F; \\ \varepsilon<v;\ \ k\in K;\ \ l,f\in L;\ \ f<l;\ \ r,j\in J:(k,r,j)\in R, \tag{11}$$

$$lq_{ijkl} \geqslant \rho_k - (H+1)\left(1-q_{ijkl}\right); \quad i\in I;\ \ l\in L;\ \ (k,j)\in O: k\in K^2, \tag{12}$$

$$iq_{ijkl} \geqslant \tau q_{\tau rkf} - (H+1)\left(1-q_{ijkl}\right); \quad \tau\in I;\ \ (k,r,j)\in R;\ \ f,l\in L, \tag{13}$$

$$lq_{ijkl} - fq_{\tau rkf} \leqslant p_{jk} + p_{rk} - 1 - (H+1)\left(2 - q_{ijkl} - q_{\tau rkf}\right); \\ (k,r,j)\in R;\ \ i\in I;\ \ f,l\in L;\ \ f<l, \tag{14}$$

$$z_k = \sum_{i\in I}\sum_{l\in L}\frac{lq_{ijkl}}{p_{jk}} + \frac{p_{jk}-1}{2}; \quad (k,j)\in P;\ \ p_{jk}>0, \tag{15}$$

$$e_k^1 \geqslant z_k - t_k^1\,; \qquad k\in K^2, \tag{16}$$

$$e_k^2 \geqslant t_k^1 - z_k\,; \qquad k\in K^2, \tag{17}$$

$$z_k - t_k^2 \leqslant (H+1)w_k\,; \qquad k\in K^2, \tag{18}$$

$$lq_{ijkl} - fq_{\tau rkf} - 1 \leqslant g_{\varepsilon v} + p_{rk} + \sum_{\delta \in L} y_{vk\delta} + (H+1)\left(1 - q_{\tau rkf}\right);$$
$$(i,v),\ (\tau,\varepsilon) \in F;\ \ \varepsilon < v;\ \ f,l \in L;\ \ f < l;\ \ (k,r,j) \in R, \qquad (19)$$

$$ly_{vkl} \geqslant \sum_{f \in L} \sum_{\tau \in I:\ (\tau,\varepsilon) \in v} \frac{fq_{\tau rkf}}{p_{rk}} + \frac{p_{rk}+1}{2} + g_{\varepsilon v} - (H+1)\left(1 - y_{vkl}\right);$$
$$(k,r,j) \in R;\ \ l \in L;\ \ \varepsilon, v \in V;\ \ \varepsilon < v;\ \ p_{jk} > 0;\ \ p_{rk} > 0, \qquad (20)$$

$$ly_{vkl} \leqslant \sum_{f \in L} \sum_{i \in I:\ (i,v) \in F} \frac{fq_{ijkf}}{p_{jk}} - \frac{p_{jk}+1}{2} + (H+1)\left(1 - y_{vkl}\right);$$
$$(k,r,j) \in R;\ \ v \in V;\ \ v > 1;\ \ l \in L;\ \ p_{rk} > 0, \qquad (21)$$

$$\sum_{k \in K} y_{vkl} \leqslant d_v; \qquad l \in L;\ \ v \in V;\ \ v > 1, \qquad (22)$$

$$e_k^1,\ e_k^2,\ z_k \geqslant 0; \qquad k \in K^2, \qquad (23)$$

$$y_{vkl},\ q_{ijkl},\ w_k \in \{0,\ 1\};\ \ i \in I;\ \ k \in K;\ \ v \in V;\ \ l \in L. \qquad (24)$$

The presented mathematical model is intended for determining schedules with the best possible timeliness of order completion. This is achieved by minimizing the costs (1) associated with delayed or premature completion of products. The constraints concerning the model M ensure: (2) – allocation of all operations among the production machines (3) – the possibility of making the products specified by the decision-maker in accordance with the original schedule; (4) – elimination of assigning the operations to unsuitable machines; (5) – execution of no more than one operation for each product in any given period; (6) – execution of no more than one operation at a time by the production machine, if in this period the machine is available for operation executing; (7) continuity of performance of each operation (8) – no more than one machine at each stage being loaded by the given product; (9) – indivisibility of operations between various machines; (10) – operations performed in accordance with given limitations regarding their order; (11) – reserving time for transport of the products between stages; (12) – execution the products only when the production system is ready for this; (13) – unidirectional product flow along the production system; (14) – performance of various successive operations on one machine, concerning the same product; (15) – determining the time of completing the product; (16) – determining delay of completing the product if the deadline for product completion has lapsed; (17) – determining the time for which the product must be stored due to premature completion of the operations regarding this product; (18) – deciding about imposing a fine for missing the latest deadline for product completion. Further constraints regard usage of intermediate buffers, and ensure: (19) – determination of time when the given product will stay in the buffer located before the specified stage; (20) – locating the product in the buffer after completing the preceding operation and transporting it from the last loaded machine; (21) – locating the product in the buffer before performing next operation; (22) – providing for a limited number of intermediate buffers. Constraints (23) and (24) ensure appropriate types of variables.

3. The relaxation heuristic

The mathematical model presented in the previous section was used in the relaxation heuristics. Relaxation of described models was performed. The binary variables were replaced with continuous variables. Next, the rounding rules were applied, along with verification and modification procedures for the resulting solution. Tasks of significant sizes can be solved in a relatively short time using described below heuristic.

The variables special defined for the iterations of heuristic, where a – the number of iteration:

$\widetilde{q}^{a}_{ijkl} = 1$ if product k is assigned to machine i to perform operation j in period l, otherwise $\widetilde{q}^{a}_{ijkl} = 0$;

s^{a}_{ijk} – time of starting operation j on machine i regarding product k;

z^{a}_{ijk} – time of ending operation j on machine i regarding product k.

These times are the final solution of the rescheduling task.

The relaxation heuristic H:

Step 1. Linear relaxation of the M mathematical model and preliminary allocation of the operations to the production machines.

Apply the mathematical model M, omitting constraint (24), and declare nonzero of all the decision variables, that is solve the dependence system (1)–(23).

The algorithm retains assignments of operations between machines for products $k \in K^1$, which are to be made in accordance with the original schedule. This was taken into account in step 3. For products $k \in K^2$, machines are assigned in further steps of the heuristic.

Allocations of the operations to the machines for individual products will be determined and stored thanks to variables $x_{ijk} \in \{0, 1\}$, where $x_{ijk} = 1$, if product $k \in K^2$ is assigned to machine i to perform operation j. Clear the values of these variables, i.e. assume $x_{ijk} = 0$ for $i \in I$, $j \in J$, $k \in K^2$, and perform the following:

a) Assume $k := 0$ (the index of the type of product), and go to step 1b.

b) If $k < W$ (W – the number of product types), than assume: $k := k + 1$, $j := 0$, and go to step 1c; otherwise, go to step 2.

c) If $k \in K^1$ then return to step 1b; otherwise, assume $j := j + 1$, go to step 1d.

d) If $j > N$ (N – the number of operation types), than go to step 1b. If $(k, j) \notin O$, return to step 1c.

Determine allocations of the operations to the machines as follows: assign $x_{ijk} = 1$ to this machine i able to perform operations j, for which the value $\sum_{l \in L} q_{ijkl}$ is the highest. If the value of this sum is identical for several machines, choose a machine able to perform operation j with a lower index i. Go to step 1c.

Step 2. Verification and modification of the preliminary allocation of the operations to the machines.

This step verifies if the product $k \in K^2$ loads no more than one machine at the given stage. Assume $k := 0$ and $v := 0$ (the number of stage), proceed to step 2a.

a) Assume $k := k + 1$. If $k > W$ (Y – the number of product types), go to step 3; otherwise, go to step 2b.

b) If $k \in K^1$ then return to step 2a; otherwise, assume $v := v + 1$. If $v > A$ (A – the number of stages), return to step 2a; otherwise, go to step 2c.

c) If constraint (25) is fulfilled, return to step 2b.

$$x_{ijk} + x_{\tau rk} \leqslant 1\,; \qquad r, j \in J;\ \ r \neq j;\ \ (i, v),\ (\tau, v) \in F;\ \ \tau \neq i\,. \tag{25}$$

If the constraint (25) is not met, choose, from among machines i located at stage v, one machine, such that $(i, v) \in F$ as per the following lexicographical order:

- the machine of the largest value of the sum $\sum_{j \in J} \sum_{l \in L} q_{ijkl}$;
- the machine of the least loading – machine of the smallest value $\sum_{j \in J} \sum_{l \in L} \sum_{r \in K} q_{ijlr}$;
- the machine of the smallest index i, and $(i,\ v) \in F$.

Mark the selected machine as i', and next mark the set of operations concerning the product k performed at stage v as J'. Then, verify the allocation of the operations to the machines using (26), and next return to step 2b.

$$x_{ijk} = \begin{cases} 1 & \text{for } i = i',\ j \in J' : (i, v) \in F, \\ 0 & \text{for } i \neq i',\ j \in J' :\ (i, v) \in F. \end{cases} \tag{26}$$

Step 3. Distribution of the operations in time and consideration of constraints connected with the original schedule.

Assume iteration number $a := 1$ and round the part off the decision variables (for products $k \in K^2$) to the nearest integer. Store the rounded-off values using variables $\widetilde{q}^{a}_{ijkl}$ using (27); round – the rounding function.

$$\widetilde{q}^{a}_{ijkl} = \begin{cases} 1; & i \in I;\ \ j \in J;\ \ k \in K;\ \ l \in L\colon (i, j, k, l) \in Z, \\ \text{round}\left(q_{ijkl}\right); & i \in I;\ \ j \in J;\ \ k \in K^2;\ \ l \in L. \end{cases} \tag{27}$$

For products $k \in K^1$, operations are still assigned to machines in accordance with the original schedule.

Determine the starting times for individual operations on the basis of (28), and the ending times based on (29).

$$s^{a}_{ijk} = \begin{cases} \displaystyle\sum_{l \in L} \frac{l\widetilde{q}^{a}_{ijkl}}{p_{jk}} - \frac{p_{jk} - 1}{2}; & i \in I;\ \ j \in J;\ \ k \in K^1;\ \ p_{jk} > 0, \\ x_{ijk} \cdot \min\limits_{l \in L} \left(l\widetilde{q}^{a}_{ijkl}\right); & i \in I;\ \ (k, j) \in O; \end{cases} \tag{28}$$

$$z^{a}_{ijk} = \begin{cases} \displaystyle\sum_{l \in L} \frac{l\widetilde{q}^{a}_{ijkl}}{p_{jk}} + \frac{p_{jk} - 1}{2}; & i \in I;\ \ j \in J;\ \ k \in K^1;\ \ p_{jk} > 0, \\ s^{a}_{ijk} + p_{jk} - 1; & i \in I;\ \ (k, j) \in O. \end{cases} \tag{29}$$

Assume $a := a + 1$ and store the loads of individual machines in time intervals l, using (30); proceed to step 4.

$$\widetilde{q}^{a}_{ijkl} = \begin{cases} 1, & \text{if } s^{a-1}_{ijk} \geqslant l \leqslant z^{a-1}_{ijk} \\ 0, & \text{otherwise} \end{cases};\quad i \in I;\ \ j \in J;\ \ k \in K;\ \ l \in L. \tag{30}$$

Step 4. Verification of intervals between operations performed on the same machine regarding the same product.

a) Check the constraint (31) for every pair of consecutive operations $(r,\ j)$ concerning product $k \in K^2$, where $(k, r, j) \in R$ performed on the same production machine ($x_{ijk} = x_{irk} = 1$). If the constraint (31) is met, than every two consecutive operations concerning the product, performed on the same machine, are carried out directly one after the other. In such a case, go to step 5, otherwise proceed to step 4b.

$$s^a_{irk} - z^a_{ijk} = 1; \quad i \in I; \ (k, r, j) \in R; \ x_{ijk} = x_{irs} = 1. \tag{31}$$

b) Assume $a := a + 1$. Mark every product k' which fails to meet constraint (31) in the case of machine i' (mark it like that), and ten determine the set of operations performed on this machine and mark it as J'. Next the first operation performed on production machine i' concerning product s' is to be marked as j'. The starting and ending times for these operations, for which equation (31) is observed, do not change – they are the same as in iteration $a - 1$. Those times are determined in iteration a based on constraints (32) and (33). Whereas for operations and products which failed to meet the constraint (31), the equations (34) and (35) are used to determine such operation starting and ending times as to meet the constraint (31).

$$s^a_{ijk} = s^{a-1}_{ijk}; \quad k \in K^2; \ (i \in I \backslash \{i'\}; \ j \in J) \vee (i = i'; \ j \in J \backslash J' \vee j = j'), \tag{32}$$

$$z^a_{ijk} = s^a_{ijk} + p_{jk} - 1; \quad k \in K^2; \ (i \in I \backslash \{i'\}; \ j \in J) \vee \\ \vee (i = i'; \ j \in J \backslash J' \vee j = j'), \tag{33}$$

$$s^a_{ijk} = z^a_{irk} + 1; \quad i = i'; \ r \in J'; \ j \in J' \backslash \{j'\}; \ (k', r, j) \in R, \tag{34}$$

$$z^a_{ijk} = s^a_{ijk} + p_{jk} - 1; \quad i = i'; \ j \in J' \backslash \{j't\}; \ k \in K^2; \ k = k'. \tag{35}$$

Constraints (34) and (35) are to ensure continuous performance of operations assigned to the same product and on one machine. Update, using equestion (36), the information on machine loading in time intervals. Next cancel emphasis of machines, products, production operations and operation set (i', j', k', J'), and proceed to step 5.

$$\tilde{q}^a_{ijkl} = \begin{cases} 1, & \text{if } s^a_{ijk} \geqslant l \leqslant z^a_{ijk} \\ 0, & \text{otherwise} \end{cases}; \quad i \in I; \ j \in J; \ l \in L; \ k \in K^2. \tag{36}$$

Step 5. Verification of collision-free performance of the production operations.

At this step, check whether in the given time interval l every machine i performs no more than one operation.

a) Assume $i := 0$, and proceed to step 5b.

b) Assume $i := i + 1$ and $l := 0$, and next proceed to step 5c.

c) Assume $l := l + 1$. If constraint (37) is met, ensuring that in the given time period l the machine i performs no more than one production operation, go to step 6; otherwise, proceed to step 5d.

$$\sum_{j\in J}\sum_{k\in K}\widetilde{q}^{a}_{ijkl} \leqslant n_{il}\,. \tag{37}$$

d) As constraint (37) is not met for the time interval l and the machine i, choose only one product k, which meets the condition $\widetilde{q}^{a}_{ijkl} = 1$ ($j \in J$). Next mark the selected product as k'. Choose product k' using the lexicographical order:

1. Product $k \in K^1$.
2. Product which in the previous iteration met the equation $\widetilde{q}^{a}_{ijkl} = 1$ ($j \in J$) and was not marked as k'.
3. The product of the smallest nonzero value of variable s^{a}_{ijk} ($j \in J$);
4. The product characterized of the largest value of the sum: $\sum_{j\in J}\sum_{l\in L}\widetilde{q}^{a}_{ijkl}$;
5. The product of the smallest index k.

Let j' mean an operation concerning product k' performed in the time interval l in production machine i. Assume $a := a+1$, and using (38), determine the number of time intervals b, where the operation j' requires the production machine i to be loaded in the period between l and the end of this operation.

$$b = c^{e-1}_{ijs'} - l + 1. \tag{38}$$

Applying condition (39), determine the elements of the set O' – a set of ordered pairs (k, j), where the operation j concerns the product k. The set O' includes products and their respective operations for which the schedule must be modified – "right shifts" by the designated value b of the time intervals.

In order to ensure that the production is collision-free, modify the schedule using (40). Next update the operation starting and finishing times for products using the equations (28) and (29). Cancel emphasis of operations and products (j', k') and proceed to step 6.

$$O' = \left\{(k, j)\colon k \in K^2;\;\; j \in J\backslash\{j'\}\colon \widetilde{q}^{a-1}_{ijkf} = 1;\;\; f \in \langle l, l + b - 1\rangle\right\}, \tag{39}$$

$$\widetilde{q}^{a}_{\tau jk\rho} = \begin{cases} \widetilde{q}^{a-1}_{\tau jk\rho} & \text{for } \left((\tau \in I\backslash\{i\};\ (k,j) \in O') \vee \right. \\ & \left. \vee\ (\tau = i;\ j \in J;\ (k,j) \notin O') \right) \\ \widetilde{q}^{a-1}_{\tau jkf} & \text{for } \tau = i;\ (k,j) \in O';\ \rho \geqslant l+b,\ f = \rho-b \end{cases} ;\quad \begin{matrix} \rho \in L; \\ k \in K. \end{matrix} \tag{40}$$

Step 6. Verification of compliance with constraints concerning the order of the operations.

If the verified constraint (41) concerning order restrictions is met for the operation j performed on machine i and its predecessor – operation r performed on machine τ, then go to step 7; otherwise, mark the operation which fails to meet the constraint (41), as j', and mark the product which this operation concerns as k'. Next determine b – the minimum number of time intervals by which the value of $s^{a}_{ij'k'}$ is to be increased, for the constraint (41) to be observed.

$$s^{a}_{ijk} - z^{a}_{\tau rk} - 1 \geqslant g_{\varepsilon v}]; \qquad (i,v),\ (\tau,\varepsilon) \in F;\quad (k,r,j) \in R;\quad s^{a}_{ijk} \leqslant l. \tag{41}$$

Assume $a := a + 1$. Modify the schedule using (42). Update the operation starting and finishing times for individual products as per (28) and (29). Cancel emphasis of production operations and products (j', k'), and proceed to step 7.

$$\widetilde{q}^{a}_{\tau jk\rho} = \begin{cases} \widetilde{q}^{a-1}_{\tau jk\rho} & \text{for } ((\tau \in I\backslash\{i\};\ j \in J) \vee (\tau = i;\ j \in J\backslash\{j'\}));\\ & \qquad\qquad \rho \in L;\ k \in K, \\ \widetilde{q}^{a-1}_{\tau jkf} & \text{for } \tau = i;\ j = j';\ \rho \in L;\ \rho \geqslant l+b;\ f = \rho-b;\ k = k'. \end{cases} \tag{42}$$

Step 7. Halt condition for the previous phases of schedule modification.

If $l < \max\limits_{\tau\in I,\, j\in J,\, k\in K} z^{a}_{\tau jk}$, go to step 5c, if not – check relationship: $i < M$ (M – the number of machines). If the relation is fulfilled, go to step 5b; otherwise, go proceed to step 8.

Step 8. Verification of availability of intermediate buffers and verification of limited machine availability (Magiera, 2013).

At this step, the loads of each production machines and intermediate buffers in subsequent time intervals are reanalysed. The following are checked in succession:

a) availability of a intermediate buffer in time interval l, which is located before the stage v – relationship (22) is checked;

b) availability of production machine i in time interval l, on the basis on the value of the input parameter n_{il}.

For each of the presented above points the value of parameter b is determined, by which such schedule modification should be made which would ensure non-disturbance of the limited intermediate buffer capacities (point a) or availability of a production machine for loading by a given product, beginning from time interval l (point b). Before each schedule modification, the number of iteration $a := a + 1$ shall be assumed. Schedule modification of the should be made analogously to relationship (42). After each schedule modification, the starting and ending times of loading individual machines with given products should be updated using (28) and (29). These times are the final solution of the rescheduling task.

Verification of the last time interval for a production machine characterized the highest index i shall end the heuristic algorithm. Schedule length $C^h_{\max}$ is determined in accordance with equation (43).

$$C^h_{\max} = \max_{i\in I,\, j\in J, k\in K} z^a_{ijk}\,. \tag{43}$$

4. Computational experiments

The presented mathematical model and relaxation heuristic have been verified and compared by means of computational experiments. The presented mathematical model was coded in the AMPL language (*A Mathematical Programming Language*) (Fourer et al., 2003), and next *.mps files were generated, computations were performed using the GUROBI optimizer (www.gurobi.com).

The following indexes were identified:

- u_1 – intended for comparing the length of the schedule determined using heuristic H and the mathematical model M, defined in the equation (44);

$$\begin{gathered} u_1 = \frac{C^{\mathrm{H}}_{\max} - C^{\mathrm{M}}_{\max}}{C^{\mathrm{M}}_{\max}} \cdot 100\%, \quad \text{where} \\ C^{\mathrm{M}}_{\max} = \max_{i\in I,\ j\in J,\ k\in K,\ l\in L} l q_{ijkl} \quad \text{and} \quad C^{\mathrm{H}}_{\max} = \max_{i\in I,\ j\in J,\ k\in K,\ l\in L} l\, \widetilde{q}^{\,a}_{ijkl}\,, \end{gathered} \tag{44}$$

  $C^{\mathrm{M}}_{\max}$, $C^{\mathrm{H}}_{\max}$ – the length of schedule determined by means of the mathematical model M, using heuristic H; a – the number of the last iteration;

- u_2 – intended for comparing the computation times using heuristic H and the mathematical model M, defined in the equation (45);

$$u_2 = \frac{CPU^{\mathrm{H}}}{CPU^{\mathrm{M}}} \cdot 100\%, \tag{45}$$

  where CPU^{M}, CPU^{H} – computation times for the mathematical model M, the heuristic H.

The experiments were carried out for the rescheduling of machine parts production.

The computational experiments covered 5 groups of tasks. For each one of the groups, 25 examples were solved. Specification of parameters of those groups and results of computational experiments are presented in Table 2.

Table 2: Specification of parameters of groups of tasks and results of computational experiments

Group	Parameters of groups						Results of experiments	
	A	M	W	Y	N	H	u_1 [%]	u_2 [%]
1	2	4	4	8	6	14	1.9	0.36
2	3	5	4	8	8	16	2.1	0.35
3	3	6	5	10	8	28	2.9	0.30
4	3	8	5	10	10	20	2.9	0.27
5	4	8	6	12	12	22	3.1	0.25

Numbers of: A – stages, M – machines, W – types of products, Y – products, N – types of operations, H – periods

Findings of average values of results of computational experiments contained in Table 2 indicate u_2 significant reduction of computation time when using relaxation heuristic H with respect to the alternative method – integer programming mathematical model. The reduction amounted to approximately 0.25–0.36%, i.e. solutions were generated approximately 280–400 times faster than in the case of the optimal method with binary decision-making variables.

The cost of the advantage of reducing the computation times is a certain deviation from the optimal solution – this concerns the determined schedule length. This is reflected in the average values of the u_2 indexes. The schedules determined using the proposed heuristic H were longer by no more than 3.1% compared to the application of the mathematical model M.

It should also be emphasized that the heuristic method is useful mostly when large-sized problems must be solved. Frequently, it is impossible to solve such problems using discrete optimization packages, due to a limited number of variables taken into account at the same time and mathematical relations (constraints). Proposing the described heuristics is indicated in the case of need to solve problems in a relatively short time, which is often necessary in the case of rescheduling.

5. Conclusion

The main advantage of the proposed method is the possibility of very fast building of new production schedules in hybrid production lines. These schedules provide for unavailability of machines which were excluded from the production process, e.g. due to random incidents. Thanks to minimization of the costs associated with untimely completion of products, the constructed method may be positively assessed in the economic aspect. This aspect also applies to the use of buffers. The importance of buffer management in production scheduling has also been underlined in the paper of Bysko et al. (2019).

The conducted computational experiments confirmed short times of computational times using the proposed method. Yet, in the case of problems of relatively small size, when the existing do not have to be modified immediately, the presented model of integer programming task can be used. In this case, an optimum solution is determined, on account of the applied optimality criterion. Thanks to development of computer technology and software, there are very good perspectives for using integer programming in scheduling tasks of increasing size.

To sum up the directions of research regarding rescheduling, one can notice two areas of research regarding re-distribution of operations in time and space (assignment of operations to machines). One of them is construction of methods which can be used to build schedules in a short time. Another area of research concerns accurate methods. These methods are used mostly if the deadline for completing a production order is not at risk. Thanks to these methods, new, optimal schedules are built.

This paper concerns both these areas of research. The developed method concerns the first area, and the constructed mathematical model belongs to the second area of studies.

The presented mathematical model may, of course, be modified, adapted to the changing conditions of production process. E.g. if the order completion deadlines are distant, time criteria can be used: schedule length or total completion time of individual products.

Rescheduling, as shown on the example of the presented method, should be treated as a special kind of scheduling, where often additional constraints concern the original schedule, e.g. ensuring continuity of operations concerning products which were started in accordance with the original schedule. This kind of scheduling has good prospects for development. Their basis is strive for gaining competitive advantage using efficient production – with the lowest possible costs, while meeting the customers' requirements regarding product quality and compliance with the order completion deadlines.

References

[1] J. Behnamin and S.M.T. Fatemi Ghomi: Hybrid flowshop scheduling with machine and resource dependent processing times, *Applied Mathematical Modelling*, **35**(3) (2011), 1107–1123, doi: 10.1016/j.apm.2010.07.057.

[2] S. Bysko, J. Krystek, Sz. Bysko, and R. Lenort: Buffer management in solving a real sequencingproblem in the automotive industry – Paint Shop 4.0 concept, *Archives of Control Sciences*, **29**(3) (2019), 507–527, doi: 10.24425/acs.2019.130203.

[3] L. Castillo, D. Borrajo, M.A. Salido, and A. Oddi, (ed.): *Planning, scheduling and constraint satisfaction: from theory to practice*, IOS Press, Amsterdam, 2005.

[4] F.T.S. Chan, N. Li, S.H., Chung, and M. Saadat: Management of sustainable manufacturing systems – a review on mathematical problems, *International Journal of Production Research*, **55**(4) (2017), 1210–1225, doi: 10.1080/00207543.2016.1229067.

[5] R. Fourer, D.M. Gay, and B.W. Kernighan: *AMPL, A Modelling Language for Mathematical Programming*, Duxbury Press, Pacific Grove, CA, 2003.

[6] H. Furmańczyk and M. Kubale: Scheduling of unit-length jobs with bipartite incompatibility graphs on four uniform machines, *Bulletin of the Polish Academy of Sciences: Technical Sciences*, **65**(1) (2017), 29–34, doi: 10.1515/bpasts-2017-0004.

[7] T.F. Gonzales (ed.): *Handbook of Approximation Algorithms and Metaheuristics*, Chaoman and Hall/CRC, New York, 2007.

[8] K. Katragjini, E. Vallada, and R. Ruiz: Flow shop rescheduling under different types of disruption, *International Journal of Production Research*, **51**(3) (2013), 780–797, doi: 10.1080/00207543.2012.666856.

[9] J.Y.-T. Leung (ed.): *Handbook of scheduling. Algorithms, Models and Performance Analysis*, A CRC Press Company, Boca Raton, New York, Washington D.C, 2004.

[10] M. Magiera: A relaxation heuristic for scheduling flowshops with intermediate buffers, *Bulletin of the Polish Academy of Sciences: Technical Sciences* **61**(4) (2013), 929–942, doi: 10.2478/bpasts-2013-0100.

[11] M. Magiera: A multi-level method of support for management of product flow through supply chains, *Bulletin of the Polish Academy of Sciences: Technical Sciences*, **63**(4) (2015), 933–946, doi: 10.1515/bpasts-2015-0106.

[12] M.L. Pindeo: *Planning and Scheduling in Manufacturing and Services*, Springer, 2009.

[13] I. Ribas, R. Leinstein, and J.M. Framinan: Review and classification of hybrid flowshop scheduling problems from a production system and a solution procedure perspective. *Computers & Operations Research*, **37**(8) (2010), 1439–1454, doi: 10.1016/j.cor.2009.11.001.

[14] D.P. Ronconi and E.G. Birgin: Mixed-Integer Programming Models for Flowshop Scheduling Problems Minimizing the Total Earliness and Tardiness, In: Ríos-Mercado R., Ríos-Solís Y. (eds.), *Just-in-Time Systems*, Springer Optimization and Its Applications, **60**, Springer, New York, NY. 2012, doi: 10.1007/978-1-4614-1123-9_5.

[15] R. Ruiz and J.A. Vázquez-Rodriguez: The Hybrid Flow Shop Scheduling Problem, *European Journal of Operational Research*, **205**(1) (2010), 1–18, doi: 10.1016/j.ejor.2009.09.024.

[16] J.C. Seck-Tuoh-Mora, J. Medina-Marin, E.S. Martinez-Gomez, E.S. Hernadez-Gress, N. Hernadez-Romero, and V. Volpi-Leon: Cellular particle swarm optimizationwith a simple adaptive local search strategyfor the permutation flow shop scheduling problem, *Archives of Control Sciences*, **29**(2), (2019), 205–226, doi: 10.24425/acs.2019.12937.

[17] M. Soualhia, F. Khomh, and S. Tahar: Task Scheduling in Big Data Platforms: A Systematic Literature Review". *Journal of Systems and Software*, **134** (2017), 170–189.

[18] H. Suwa and H. Sandoh: Methods for Online Scheduling, In: *Online Scheduling in Manufacturing*. Springer, London, (2013), 55–73.

[19] A. Tóth, T. Knuutila, and O.S. Nevalainen: Machine configuration and workload balancing of modular placement machines in multi-product PCB assembly, *International Journal of Computer Integrated Manufacturing*, **31**(9), (2018), 815–830, doi: 10.1080/0951192X.2018.1429667.

[20] G.E. Vieira, J.W. Herrmann, and E. Lin: Rescheduling Manufaturing systems: A Framework of Strategies, Policies and Methods, *Journal of Scheduling*, **6**(1) (2003), 39–62.

[21] A. Yadov and S.C. Jayswal: Modelling of flexible manufacturing system: a review, *International Journal of Production Research*, **56**(7) (2017), 2464–2487, doi: 10.1080/00207543.2017.1387302.

[22] www.gurobi.com (Gurobi Optimizer, 2019).