DE GRUYTER OPEN

∂mper

THE OPTIMAL SEQUENCE OF PRODUCTION ORDERS, TAKING INTO ACCOUNT THE COST OF DELAYS

Robert Dylewski[1], Andrzej Jardzioch[2], Irene Krebs[3]

[1] *University of Zielona Góra, Faculty of Mathematics, Computer Science and Econometrics, Poland*
[2] *West Pomeranian University of Technology Szczecin, Faculty of Mechanical Engineering and Mechatronics, Poland*
[3] *Brandenburg University of Technology Cottbus-Senftenberg, Department of Information Systems, Germany*

Corresponding author:
Andrzej Jardzioch
West Pomeranian University of Technology Szczecin
Faculty of Mechanical Engineering and Mechatronics
Al. Piastów 19, 70-310 Szczecin, Poland
phone: (+48) 91 449-49-89
e-mail: andrzej.jardzioch@zut.edu.pl

ABSTRACT
In flexible manufacturing systems the most important element in determining the proper course of technological processes, transport and storage is the control and planning subsystem. The key planning task is to determine the optimal sequence of production orders. This paper proposes a new method of determining the optimal sequence of production orders in view of the sum of the costs related to the delayed execution of orders. It takes into account the different unit costs of delays of individual orders and the amount of allowable delays of orders involving no delay costs. The optimum sequence of orders, in the single-machine problem, in view of the sum of the costs of delays may be significantly different from the optimal order, taking into account the sum of delay times.

KEYWORDS
scheduling algorithms, production control, minimum cost of delays, single-machine processing.

Introduction

Modern flexible manufacturing systems are generally defined as integrated computer systems involving technological machines (machine tools), means of transportation, and tools enabling the effective implementation of low- and medium-volume production [1, 2]. They are characterized by a high level of integration between technological processes, transport and planning. While analysing the processes in integrated manufacturing systems, it can be assumed that the most important element in determining the proper course of technological processes, transport and storage is the control and planning subsystem [3]. One of the most important planning tasks is to determine the optimal sequence of production orders [4,5]. The criterion used to assess the individual ranking is most frequently the overall lead time of production orders, the sum of delays, the cost of delays

and the total profit derived from the realization of a set of pending orders [6–8]. The proper operation of the production system is determined by its proper cooperation with the planning subsystem. This is due to the fact that the planning subsystem handles all the important decisions concerning the transport subsystem, the storage subsystem, and the machining subsystem that have a decisive impact on the efficiency of the whole production system and the directly associated manufacturing costs. This paper presents a method of determining the optimal sequence of production orders in view of the sum of the costs related to the delayed execution of orders. One of the most important assumptions is that the single-machine problem is considered. The developed method is based on the methodology of branch and bounds, and takes into account different unit costs of delays of individual orders and the allowable delay of the order not giving rise to delay costs.

The issue of scheduling production orders, taking into account the cost of delays

The issue of scheduling production orders has been analysed in many scientific papers [9–12]. The most common scheduling criterion is the total time of execution of all orders. However, this approach is not valid in the case of modern models of production management, where logistic parameters such as deadlines, delays and penalties for any resulting delays play a vital role.

The issue of scheduling production orders has been shown on the example of the manufacturing system based on implementing single-machine processing. The considered system consists of input storage, an M1 machine and output storage. The technological process is carried out in a flow. A set of n production orders is given $Z = \{z_1, z_2, \ldots, z_n\}$. It is assumed that all orders are available at the start of production ($T_1 = 0$) and may be performed in any order. The machine can only carry out one order at a time (exclusive-like mode). Each order z_i is described by: the required processing time $t_o(i)$, the required deadline $t_t(i)$, the allowable delay of the order not causing delay costs $c(i)$ and the unit cost of delay $K_z(i)$ – the cost per unit of time for delays exceeding the value of $c(i)$. It was also assumed that the realization of individual orders requires the machine to be retooled. The machine's set-up time does not depend on the sequence of order entry to production and has been included in the processing time of individual orders. Furthermore, there are no interruptions in the machine's operation and in the delivery of orders for production.

Figure 1 shows a diagram of the issue of scheduling production orders. Input storage contains orders that have been accepted for realization. When start-ing the production process determine the sequence in which orders will be taken from the warehouse. The total number of possible order sequences is $n!$. After completion, orders end up in the output storage. Each completed order can be characterized by logistic parameters such as the actual execution time, timeliness of performance, and the amount of penalty for the delay.

The logistic parameter values of the completed orders depend on the input sequence selected. The issue of scheduling production orders comes down to determining the optimal sequence of production orders. The criterion used to assess the individual ranking is the sum of the delay costs of all orders. Based on the analysis of actual manufacturing processes, it takes into account the different unit costs of delays of individual orders and the different amounts of allowable delays of orders involving no delay costs.

The sum of the costs of delays of orders admitted for execution is a very important parameter in determining the quality of operation in a company. This parameter is related to the timely execution of production orders, but also takes into account the penalties resulting from a delay in performance. Such penalties result from a desire to protect the client against costs which might arise in the case of a delayed execution of order and should be taken into account in the production planning process. Of course, the best solution is to plan production in such a way as to make all orders on time. In this case, there are no costs associated with delays. Unfortunately, this is not always possible in commercial practice. In their struggle to obtain jobs, companies not only compete in terms of price and quality, but also the delivery date. Taking too many orders may result in a failure to implement some of them to the prearranged deadlines. In this case, it is necessary to plan production processes in order to minimize the financial impact of delays related to the realization of production orders.
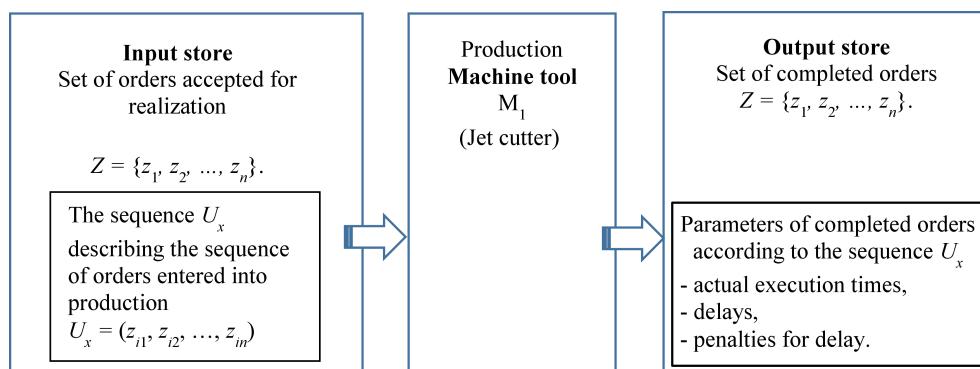

Fig. 1. The process of scheduling production orders in a single-machine production system and its impact on the logistic parameters of the produced orders

Determining the sequence of production orders with a minimum sum of the delay costs

This paper proposes a method that allows determining the sequence of production orders that allows for the minimum sum of the delay costs. The model of the production process also takes into account the so-called "allowable delay" for each order, above which one runs the risk of contractual penalties. The developed method consists of two steps. In the first stage, the "base" sequence is determined; this is the sequence for which the maximum cost associated with the delay of a single order is not greater than for all other sequences. The second stage defines the sequence of orders, ensuring the minimum sum of the cost of delays.

It is assumed that for a given list of orders $z_1, z_2, ..., z_n$ the following information is available:

$t_o(i)$ – the processing time for order z_i,

$t_t(i)$ – the required deadline for order z_i,

$c(i)$ – the allowable delay for order z_i not causing delay costs,

$K_z(i)$ – the unit cost of delay for order z_i (cost per unit of delay time exceeding the value $c(i)$).

The process of determining the base sequence begins by determining the sum of the processing times of all orders received. S_o marks the sum of processing times in the given list of n orders:

$$S_o = \sum_{i=1}^{n} t_o(i). \qquad (1)$$

For each job in the list it is verified whether implementing it as the last one would result in a delay, and whether it will be necessary to pay the penalty. If order z_i is executed as the last, its due date will be equal to the sum of processing times of all orders. The cost associated with the delay in the execution of this order shall be:

$$p(i) = \max \left\{ (S_o - t_t(i) - c(i)) \cdot K_z(i); 0 \right\}. \qquad (2)$$

The introduction of the parameter $c(i)$ describing the acceptable delay allows a better adaptation of the model of the issue at hand to the requirements found in the planning practice of industrial plants. Based on the analysis of production processes, it can be noted that small delays in the execution of production orders often do not result in any penalties.

Mathematically speaking, parameter $c(i)$ is equivalent to introducing a modified deadline $t_t'(i)$ for the execution of order z_i:

$$t_t'(i) = t_t(i) + c(i). \qquad (3)$$

In the case where all $c(i) = 0$, and all $K_z(i) = 1$, the issue comes down to designating a sequence that provides the minimum sum of delay times. Then $p(i) = \max \left\{ S_o - t_t(i); 0 \right\}$ and means a delay for the execution of order z_i, should it be carried out last. This problem has been presented in detail in the authors' previous paper [8].

Determining solutions with a minimum sum of the costs of delays is in the form of a tree.

Stage I:

Initially, in the first block (the root of the tree), the value of $p(i)$ is determined for each task according to the formula (2). If $p(i) = 0$ for every $i = 1, ..., n$, this means that each sequence of production orders (of all possible options) gives the sum of the delay costs equal to 0. In this case, the process of finding the optimal solution is completed. In the event that not all $p(i)$ are equal to 0, you select any of the orders for which $p(i)$ is the smallest and place it at the end of the queue (mark the selected order as z_j). Then you determine the time S_o' that is needed to carry out the rest of the orders (excluding the selected order z_j); this equals

$$S_o' := S_o - t_o(j). \qquad (4)$$

The total cost of delays (for the time being, only taking into account the last order in the queue) is $K_{op} = p(j)$.

Then, in block 2 (the successor of block 1 in the tree of solutions), it is performed what has already been done at the root of the tree, but without the order that has already been put at the end of the queue. Wherein, now to determine $p(i)$ you take into account $S_o' (p(i) = \max \left\{ (S_o' - t_t(i) - c(i)) \cdot K_z(i); 0 \right\})$. The order selected in block 2 is designated as z_k, and inserted into the queue as the second last. The time required for processing the remaining orders $S_o' := S_o' - t_o(k)$ and the sum of delay costs $K_{op} := K_{op} + p(k)$.

Thereafter, repeat what has been done in block 2 (ignoring the orders already entered as the last and second last in the queue) until you reach block of n (leaf in the tree), from which the order is set to the front of the queue. This way you have obtained the base branch in the tree (with the established sequence of all orders). If the sum of the cost of delays for the entire branch SK_{op} is 0, the resulting sequence is optimal in terms of the sum of the delay costs. The minimum sum of delay costs in this case is 0.

Stage II:

If the sequence determined in the base branch has $SK_{op} > 0$, the sequence does not necessarily give the minimum total cost of delays. Check from block $n-1$ to block 1, whether selecting the next or-

der (in terms of its minimum value $p(i)$), would exceed the sum of delay costs SK_{op} of the basic solution. If so, do not expand another branch. If not (i.e. $K_{op} + p(i_k) <= SK_{op}$), enter the selected order z_{ik} into the next block in the appropriate place in the queue, and so on.

If you reach the next leaf in the tree (with a fixed sequence of all orders), then the sum of delay costs for this sequence is not higher than for the basic solution. Update SK_{op} and continue checking subsequent branches, while it is possible to expand some (until $SKop$ is exceeded). Finally, optimal solutions are the leaves with the minimum SK_{op} value; denote it by SK_{opm}.

Comments:

Denote the fixed sequence of all orders as $(z_{i1}, z_{i2}, \ldots, z_{ik}, \ldots, z_{in})$. If a block with the established sequence of $n-k$ orders for all other orders shows $p(i) = 0$, then the sum of the delay costs SK_{op} for the entire branch will be the same as the sum of the delay costs K_{op} in that block. Therefore, the sequence of the remaining orders in the primary k positions is then unrestricted. Such a situation should be marked with $([z_{i1}, z_{i2}, \ldots, z_{ik}], \ldots, z_{in})$. In particular, when in the first block $p(i) = 0$ for all $i = 1, \ldots, n$, then we have $([z_1, z_2, \ldots, z_n])$.

Examples

To better understand the proposed method, consider two examples. Data for Example 1 are shown in Table 1.

Variants differ by: $ci)$ – the allowable delay not resulting in the cost of delays being charged and $K_z(i)$ – the unit cost of delay. Note that in variant 1, the total cost of delay will be equal to the sum of delay times. In variants V2–V4, acceptable delays were set for all orders at 10, 20 and 30 min, respectively. Variant V5 differs from V2 in that the unit cost of an order delay z_3 is 10 times larger than for the others.

Table 2 summarizes the results. Sequences for base solutions and optimal solutions were given for all variants. Note that the inclusion of parameters $c(i)$ and $K_z(i)$ has a significant impact on what the optimal sequence will be in terms of the sum of the costs of delays. Moreover, this sequence may be significantly different from optimal due to the sum of the delay times (see variants V1 and V5).

The figures below show trees with solutions for variants of Example 1. In the upper left corner, the block number according to the designation sequence is given under (Bi). The crossed out order zi indicates that attaching this order to the item would exceed the minimum sum of the delay costs, i.e. $K_{op} + p(i) > SK_{op}$.

Table 1
Data for Example 1.

Order	Processing time t_o [min]	Required deadline t_t [min]	Variant V1 c/K_z	Variant V2 c/K_z	Variant V3 c/K_z	Variant V4 c/K_z	Variant V5 c/K_z
z_1	10	150	0/1	10/1	20/1	30/1	10/1
z_2	20	30	0/1	10/1	20/1	30/1	10/1
z_3	100	110	0/1	10/1	20/1	30/1	10/10
z_4	50	60	0/1	10/1	20/1	30/1	10/1

Table 2
Summary of the designated sequences selected for Example 1.

Ident.	Sequence	SK_{op} (V1)	SK_{op} (V2)	SK_{op} (V3)	SK_{op} (V4)	SK_{op} (V5)	Comments
U_1	(z_2, z_4, z_3, z_1)	0+10+60+30=100	0+0+50+20=70	0+0+40+10=50	0+0+30+0=30	0+0+50*10+20=520	is base for V1, V2, V3, V4 and optimal for V3, V4
U_2	(z_2, z_4, z_1, z_3)	0+10+0+70=80	0+0+0+60=60	0+0+0+50=50	0+0+0+40=40	0+0+0+60*10=600	is optimal for V1, V2, V3
U_3	$([z_1, z_2], z_4, z_3)$	0+0+20+70=90	0+0+10+60=70	0+0+0+50=50	0+0+0+40=40	0+0+10+60*10=610	is optimal for V3
U_4	(z_2, z_3, z_4, z_1)	0+10+110+30=150	0+0+100+20=120	0+0+90+10=100	0+0+80+0=80	0+0+100+20=120	is base for V5
U_5	(z_2, z_3, z_1, z_4)	0+10+0+120=130	0+0+0+110=110	0+0+0+100=100	0+0+0+90=90	0+0+0+110=110	is optimal for V5

Figure 2 shows the tree of solutions for variant V1 for example 1. The block B4 has a solution with the base sequence which proved not to be optimal. The minimum sum of the cost of delays has been obtained in block B7. The crossed out order z_2 from block B3 means that $(_{Ko}p + p(2) = 90 + 40 = 130) > (S_{Ko}p = 100)$, i.e. insert z_2 at the third position from the end, before z_3, z_1 will make the total cost of delays at least 130, which is more than for the base sequence. The same is true with other strikethroughs.

Figure 3 shows the tree of solutions for variant V2 from example 1. Variant V2 is different from V1 in that for each order the acceptable delay without the delay costs is 10 min (in case of V1 – 0 min). The resulting tree has the same structure as in the case of V1, and hence the base sequence and the optimal sequence are the same. The value of the minimum sum of the delay costs is smaller in variant V2.
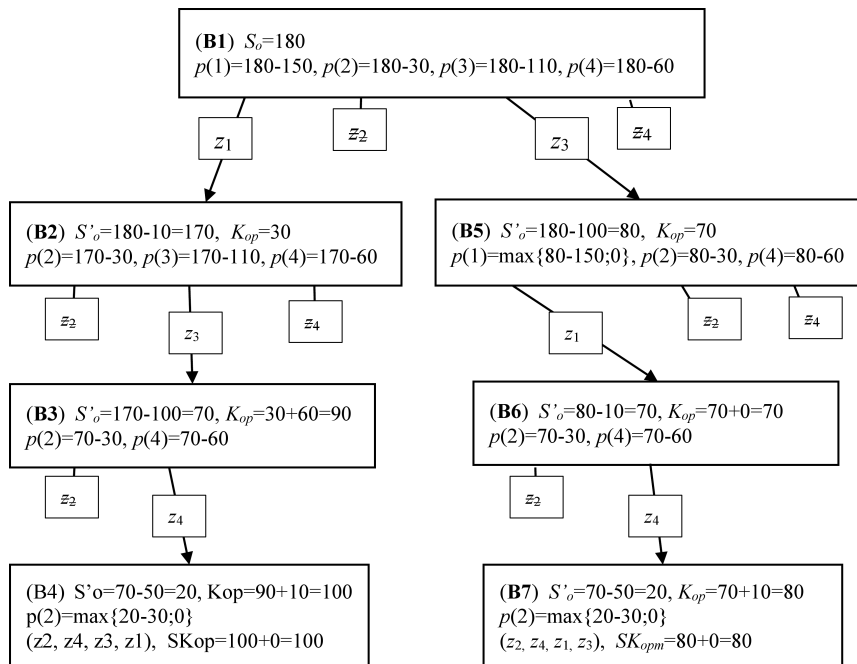
(B1) $S_o=180$
$p(1)=180-150$, $p(2)=180-30$, $p(3)=180-110$, $p(4)=180-60$

z_1 $\not{z_2}$ z_3 $\not{z_4}$

(B2) $S'_o=180-10=170$, $K_{op}=30$
$p(2)=170-30$, $p(3)=170-110$, $p(4)=170-60$

(B5) $S'_o=180-100=80$, $K_{op}=70$
$p(1)=\max\{80-150;0\}$, $p(2)=80-30$, $p(4)=80-60$

$\not{z_2}$ z_3 $\not{z_4}$ z_1 $\not{z_2}$ $\not{z_4}$

(B3) $S'_o=170-100=70$, $K_{op}=30+60=90$
$p(2)=70-30$, $p(4)=70-60$

(B6) $S'_o=80-10=70$, $K_{op}=70+0=70$
$p(2)=70-30$, $p(4)=70-60$

$\not{z_2}$ z_4 $\not{z_2}$ z_4

(B4) S'o=70-50=20, Kop=90+10=100
p(2)=max{20-30;0}
(z2, z4, z3, z1), SKop=100+0=100

(B7) $S'_o=70-50=20$, $K_{op}=70+10=80$
$p(2)=\max\{20-30;0\}$
(z_2, z_4, z_1, z_3), $SK_{opm}=80+0=80$

Fig. 2. Tree of solutions for variant V1 for example 1.

(B1) $S_o=180$
$p(1)=180-160$, $p(2)=180-40$, $p(3)=180-120$, $p(4)=180-70$

z_1 $\not{z_2}$ z_3 $\not{z_4}$

(B2) $S'_o=180-10=170$, $K_{op}=20$
$p(2)=170-40$, $p(3)=170-120$, $p(4)=170-70$

(B5) $S'_o=180-100=80$, $K_{op}=60$
$p(1)=\max\{80-160;0\}$, $p(2)=80-40$, $p(4)=80-70$

$\not{z_2}$ z_3 $\not{z_4}$ z_1 $\not{z_2}$ $\not{z_4}$

(B3) $S'_o=170-100=70$, $K_{op}=20+50=70$
$p(2)=70-40$, $p(4)=70-70$

(B6) $S'_o=80-10=70$, $K_{op}=60+0=60$
$p(2)=70-40$, $p(4)=70-70$

$\not{z_2}$ z_4 $\not{z_2}$ z_4

(B4) $S'_o=70-50=20$, $K_{op}=70+0=70$
$p(2)=\max\{20-40;0\}$
(z_2, z_4, z_3, z_1), $SK_{op}=70+0=70$

(B7) $S'_o=70-50=20$, $K_{op}=60+0=60$
$p(2)=\max\{20-40;0\}$
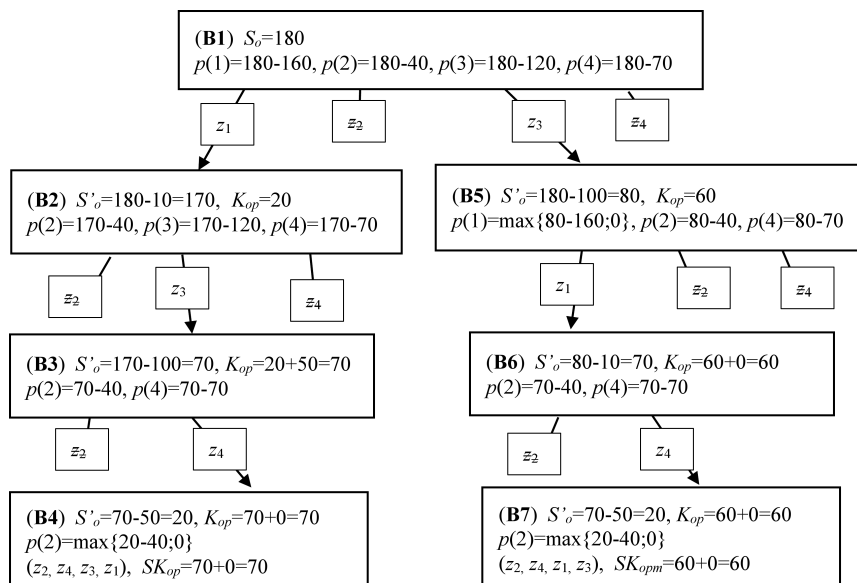(z_2, z_4, z_1, z_3), $SK_{opm}=60+0=60$

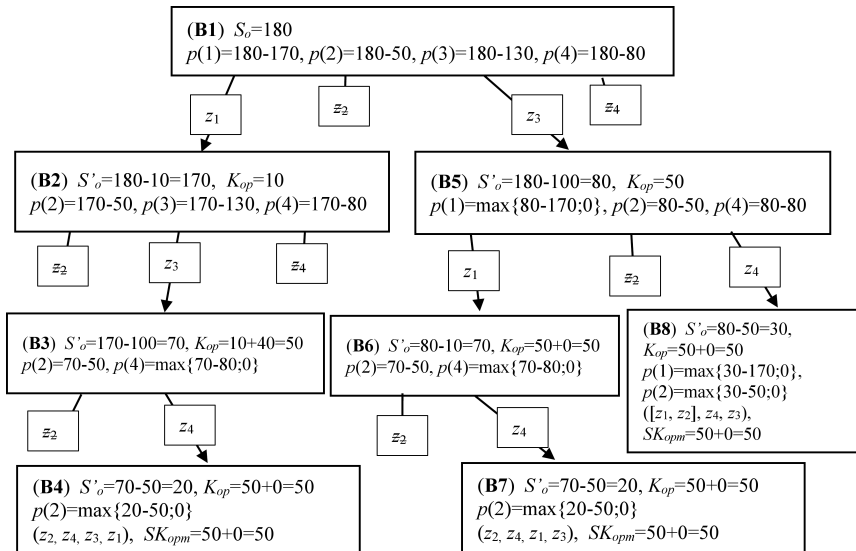Fig. 3. Tree of solutions for variant V2 for example 1.

Fig. 4. Tree of solutions for variant V3 for example 1.

Figure 4 shows the tree of solutions for variant V3 for example 1. This variant assumes a greater acceptable delay without charging the cost of delays; 20 min for each order. In this case, the basic solution is the same as in the case of V1, in block B4. This solution is also optimum for this variant. Moreover, there are three other optimal solutions, in blocks: B7 (z_2, z_4, z_1, z_3) and B8 (z_1, z_2, z_4, z_3) and z_2, z_1, z_4, z_3).

Figure 5 shows the tree of solutions for variant V4 for example 1.
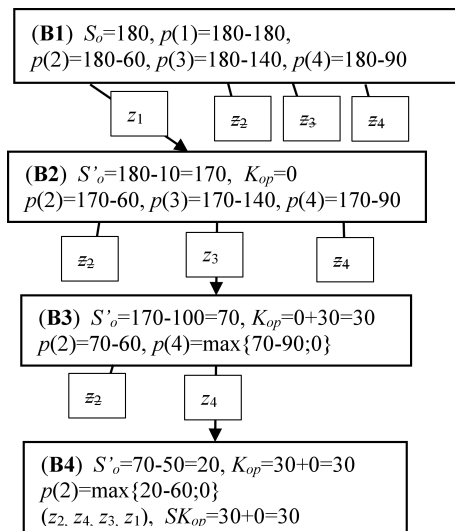

Fig. 5. Tree of solutions for variant V4 for example 1.

Even larger values (30 min) of the acceptable delay not resulting in delay costs were accepted for all orders. This time the tree has the simplest structure. The resulting basic solution in block B4 is also the only optimal solution

The last Fig. 6 shows the tree of solutions for variant V5 for example 1. As it has been mentioned previously, it differs from variant V2 only in that the unit cost of delay for order z_3 is 10 instead of 1. Unfortunately, in this case the basic solution differs significantly from the base of the previous variants. Similarly, the optimal solution obtained in block B7 is different from that in variant V2. The minimum total cost of delays is $S_{Kopm} = 110$. Whereas for the optimal solution in variant V2(z_2, z_4, z_1, z_3), the total cost of delays in variant V5 would be as much as 600.

The second example, the data for which is given in Table 3, is also considered. The presented variants are differentiated by $K_z(i)$ – the unit cost of delay. Similarly as in Example 1, variant V1, the total costs of delay equal the sum of delay times. In variant V2, the unit costs of delay for orders z_3, z_4 and z_5 are greater than 1.

Table 4 shows the results obtained for this example. Sequences for base solutions and optimal solutions were given for both variants.
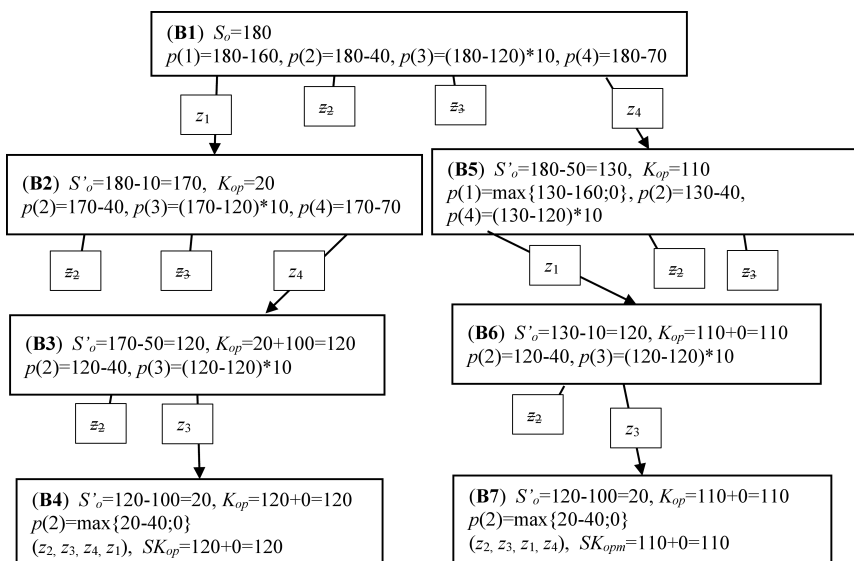
Fig. 6. Tree of solutions for variant V5 for example 1.

Table 3
Data for Example 2.

Order	Processing time t_o [min]	Required deadline t_t [min]	Variant V1 c/K_z	Variant V2 c/K_z
z_1	20	30	0/1	0/1
z_2	40	60	0/1	0/1
z_3	30	70	0/1	0/2
z_4	50	80	0/1	0/2
z_5	60	120	0/1	0/4

Table 4
Summary of the designated sequences selected for Example 2.

Ident.	Sequence	SK_{op} (V1)	SK_{op} (V2)	Comments
U_1	$(z_1, z_2, z_3, z_4, z_5)$	0+0+20+60+80=160	0+0+20*2+60*2+80*4=480	is base for V1 and optimal for V1
U_2	$(z_3, z_4, z_5, z_1, z_2)$	0+0+20+130+140=290	0+0+20*4+130+140=350	is base for V2
U_3	$(z_1, z_3, z_5, z_2, z_4)$	0+0+0+90+120=210	0+0+0+90+120*2=330	is optimal for V2

It is noteworthy that the inclusion of parameter $K_z(i)$ has a significant impact on what the base sequence and the optimal sequence will be in terms of the sum of the costs of delays. Both the base sequence and the optimal sequence may be significantly different from the base and optimum in terms of the sum of delay times, as shown in this example.

Conclusions

The problem becomes significantly more complicated when, apart from processing times and the required deadlines, the determination of the costs of delays takes into account two additional parameters (the sum of acceptable delay not resulting in delay costs, and the unit cost of order delays). The proposed method allows determining the optimum solution in terms of the minimal amount of delay costs, taking into account these two parameters.

The examples presented in the paper demonstrate that a mere adding of $c(i) = c = $ const (acceptable delay for order z_i that would not result in charging delay costs, the same for all orders) may result in the sequence that is better in terms of the sum of delay times being worse in terms of the sum of delays costs, and vice versa (although the base solutions are the same).

Taking into account the unit cost of delays of orders other than 1, means that this is the basic solution in terms of the sum of delay costs which may be different from the basic solution in terms of the sum of delay times. The optimal solution in terms of

the sum of delay costs may be all the more different from the optimum solution in terms of the sum of delay times. Further work is planned to extend the proposed approach to more complex problems (e.g. flow shop systems).

References

[1] Cheng Li, Weimin Wu, Yiping Feng, Gang Rong, *Scheduling FMS problems with heuristic search function and transition-timed Petri nets*, Journal of Manufacturing, 26, 5, 933–944, 2015.

[2] Hang Lei, Keyi Xing, Libin Han, Fuli Xiong, Zhaoqiang Ge, *Deadlock-free scheduling for flexible manufacturing systems using Petri nets and heuristic search*, Computers and Industrial Engineering, 72, 297–305, 2014.

[3] Lekurwale R.R., Akarte M.M., Raut D.N., *Framework to evaluate manufacturing capability using analytical hierarchy process*, The International Journal of Advanced Manufacturing Technology, 76, 1, 565–576, 2015.

[4] Giglio D., Minciardi R., Sacone S., Siri S., *Optimal Control of Production Processes with Variable Execution Times*, Discrete Event Dynamic Systems, 19, 3, 423–448, 2009.

[5] Yandong Guo, Min Huang, Qing Wang, Jorge L.V., *Single-machine rework rescheduling to minimize maximum waiting-times with fixed sequence of jobs and ready times,* Computers and Industrial Engineering, 91, 265–273, 2016.

[6] Lan C.-H., *The design of multiple production lines under deadline constraint*, Int. Journal of Production Economics, 106, 191–203, 2007.

[7] Kejun Zhao, Xiwen Lu, Manzhan Gu, *A new approximation algorithm for multi-agent scheduling to minimize makespan on two machines*, Journal of Scheduling, 19, 1, 21–31, 2016.

[8] Dylewski R., Jardzioch A., *Scheduling production orders, taking into account delays and waste*, Management and Production Engineering Review, 5, 3, 3–8, 2014.

[9] Lin-Hui Sun, Lin-Yan Sun, Ming-Zheng Wang, Ji-Bo Wang, *Flow shop makespan minimization scheduling with deteriorating jobs under dominating machines*, Int. J. International Journal of Production Economics, 138, 195–200, 2012.

[10] Wang J.-B., Ng C.T., Cheng T.C.E, Liu L.-L., *Single machine scheduling with a time-dependent learning effect*, International Journal of Production Economics, 111, 2, 802–811, 2008.

[11] Moshelov G. Serig A, Sidney J., *The Browne-Yechiali single machine sequence is optimal for flow-shops,* Computers and Operational Research, 37, 11, 1965–1967, 2010.

[12] Mahnama M., Moslehib G., Ghomia S.M.T.F., *Single machine scheduling with unequal release times and idle insert for minimizing the sum of maximum earliness and tardiness*, Mathematical and Computer Modelling, 57, 9–10, 2549–2563, 2013.