DE GRUYTER OPEN

Management Systems in Production Engineering

2018, Volume 26, Issue 1, pp 14-22

Date of submission of the article to the Editor: 04/2017
Date of acceptance of the article by the Editor: 10/2017

DOI 10.2478/mspe-2018-0002

MINIMIZATION OF DELAY COSTS IN THE REALIZATION OF PRODUCTION ORDERS IN TWO-MACHINE SYSTEM

Robert DYLEWSKI[1], Andrzej JARDZIOCH[2], Oliver DWORAK[1]
[1]University of Zielona Gora,
[2]West Pomeranian University of Technology

Abstract:
The article presents a new algorithm that enables the allocation of the optimal scheduling of the production orders in the two-machine system based on the minimum cost of order delays. The formulated algorithm uses the method of branch and bounds and it is a particular generalisation of the algorithm enabling for the determination of the sequence of the production orders with the minimal sum of the delays. In order to illustrate the proposed algorithm in the best way, the article contains examples accompanied by the graphical trees of solutions. The research analysing the utility of the said algorithm was conducted. The achieved results proved the usefulness of the proposed algorithm when applied to scheduling of orders. The formulated algorithm was implemented in the Matlab programme. In addition, the studies for different sets of production orders were conducted.

***Key words**: flexible manufacturing system, scheduling of orders, minimal costs of delays, two-machine flow system*

INTRODUCTION

Nowadays, in the situation of the worldwide globalisation and the increasingly fiercer competition, planning and controlling the processes of production gains special significance [6, 9, 12]. As the global economy shifts from the producer market into the direction of the consumer market, the enterprises are faced with even greater demands connected to the rationalisation and optimization of the production. This issue is particularly significant in the case of small and medium production enterprises which frequently are a part of an extended supply chain [8]. In said chain, which begins with the suppliers of the resources and finishes with the customer, the production enterprise has its own direct supplier or recipient. Proper functioning of such system requires qualitatively timely and stable production as every single disturbance in one of the links of the chain of supply may lead to the disturbance affecting the whole system [10].

Putting the absolute fulfilment of the consumer's needs in the centre of enterprise's activities and the creation of multi-stage supply chains forced a change in the priorities of the production planning. The timeliness in the realisation of orders, the minimization of the order delays, short deadlines for deliveries and low stocks gained the status of leading roles [11].

On the basis of observation of the current market, it can be stated that the situation of particular enterprises in the area of construction and technology is, in fact, quite similar. All production enterprises have access to the identical types of machines, as well as to the identical computer systems supporting the construction and development of the processing technology (CAD/CAM systems). In this situation the effective organisation of the production process and minimization of the costs connected to the possible delays in the fulfilment of the realised orders are key factors required for the achievement of the success [4].

When it comes to the proper functioning of the enterprise the problem of occurrence of delays in the realisation of the production orders is very significant. This issue is particularly essential in the case of the usage of the just-in-time management conception, in which the maximal reduction of the stock materials in the production process is implemented. Taking that into consideration, even the slightest disturbances in deliveries of the ordered goods can cause significant losses – caused by the machine stoppages – for the client. Thereafter, in order to minimise the aforementioned risk, the contracts between the supplier and the recipient contain contractual penalties for delayed fulfilment of orders. Just-in-time management conception has to organise the production in such a way that the costs of delay in the realisation of orders are eliminated, or at least, minimised.

The usage of the advanced algorithm for scheduling of the production orders is one of the effective methods enabling the minimization of the costs of the untimely realisation of orders. Such algorithm should be characterised by its operational quickness and it should also generate optimal solutions.

The article presents a new algorithm enabling the determination of the optimal scheduling of the production orders in the two-machine flow system on the account of the minimal costs connected with the realisation of the delayed orders. The designed algorithm uses the method of branch and bounds and is a particular generalisation of the algorithm enabling for the determination of the sequence of the production orders with the minimal sum of delays that was described in the research [5]. In order to illustrate the proposed algorithm in a proper manner, chapter 4 contains graphical trees of solutions. The article ends with a summary in which both the most of crucial conclusions from the conducted studies and the selected problems planned for being solved in the continuation of the research are presented.

SCHEDULING OF THE PRODUCTION ORDERS IN THE TWO-MACHINE SYSTEM

The tasks within the scope of the scheduling of the production orders are realised in the enterprises by the specialised planning cells. Systems of ERP (Enterprise Resource Planning) class and cooperating systems MES (Manufacturing Execution System) are frequently used as tools supporting the aforementioned tasks. They enable warehouse management, collection of the production information, detailed calculation and monitoring of the production costs. Despite the vast capacity of the modern computer systems supporting the planning processes, the problem of generating the production schedule in a quick way is still unsolved. There are no effective tools that could be used for the determination of the optimal sequence of the orders entered into the production system and for the proper assignment of particular orders to the machines [1].

Problem of scheduling production orders was presented on the example of an flexible manufacturing system fulfilling the production of the metal discs. The system comprises of the input storage (M_{in}), two numerically-controlled machine tools M_1 and M_2, on which the technological operations are being realised, and the output storage (M_{out}). It is the low-volume production that is being realised in the flexible manufacturing system. The aforementioned production is characterised by the occurrence of a large number of varied production orders that have clearly-defined deadlines for realisation of orders and penalties for missing said deadlines. The main task of the production planning system is the determination of the sequence of the production orders in such a way that their realisation would make the achievement of the maximal economic profits possible (Fig. 1).

In the analysed elastic system the production process is realised in a flow-like way. The realisation of each order z_i, i = 1, …, n requires the performance of two operations. Firstly, the operation $O_{1,i}$ is conducted on the machine M_1, and then the operation O_2,i is conducted on the machine M_2. Each operation $O_{j,i}$, j = 1, 2, i = 1,…, n, has the assigned time required for its realisation – $t_{j,i}$ (processing time).

The required processing time $t_{j,i}$ stems from the realised technological process and it is always positive and clearly defined. After the operation on the machine M_2 is completed, the order is passed on to the output storage. The storage capacity of the output warehouse makes it possible to store all realised orders. It was assumed that the time of setting up the machine does not depend on the order of entering the tasks into production, in addition, this time was included in the time of processing of particular orders. Moreover, there are no breaks during the work of the machine and in the delivery of the orders to the production.

The required deadline for realisation $t_t(i)$, the size of the allowed delay of the order *c(i)*, which does not result in the contractual penalty, and the unit cost of the order delay *k(i)* enabling for the calculation of the total penalty for the delay of the order, are defined for each order.

The problem of scheduling production orders in the two -machine flow system was defined as follows: it is needed to find such a sequence of entering the orders into the production U_z (scheduling of orders) which will make the achievement of the minimal costs of delay of all production orders possible. Therefore, finding particular schedule that would enable the achievement of the minimal sum of all costs caused by the delays of the production orders that exceed the limit value is seen as the goal of the scheduling process. The goal function used in the process of searching for the optimal scheduling is presented by the following formula:

$$F_c = \sum_{i=1}^{n} \max\{0, [t_r(i) - t_t(i) - c(i)] \cdot k(i)\}$$

where:

F_c – goal function used in the problem of scheduling of the production orders,

$t_r(i)$ – actual time of the realisation of the ith order,

$t_t(i)$ – required deadline for the realisation of the ith order,

c(i) – size of the allowed delay,

k(i) – unit cost of the order delay,

n – number of the production orders.

The problem which is defined in such a way belongs to NP-hard problems. The number of all possible arrangements amounts to n! and it depends on the number of the production orders. Finding the optimal solution on the basis of the total overview cannot be applied in this case as the computational complexity does not allow the achievement of the results in the accepted time (even from the 15 orders the search surface amounts to over 10^{12} permutations).

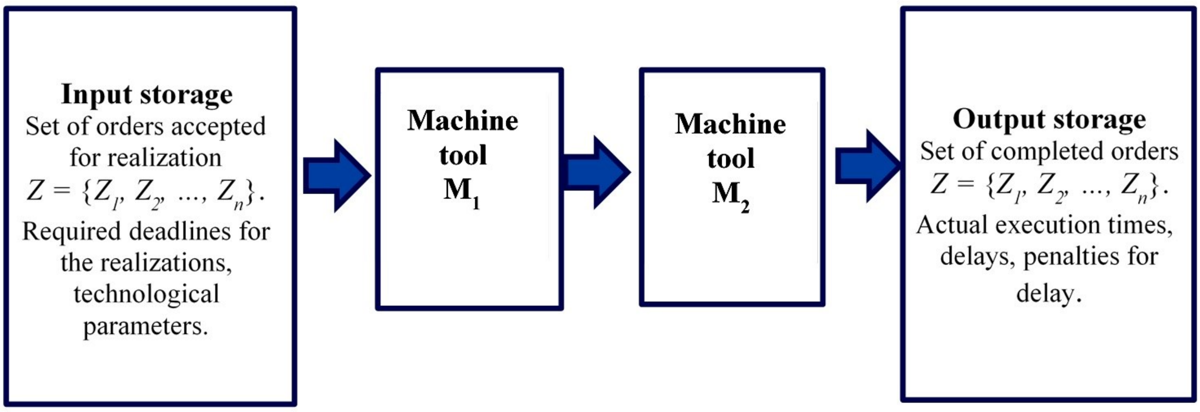

Fig. 1 Process of the order flow realised in the flexible manufacturing system

Evolutionary algorithms, clustering algorithms, algorithms of simulated annealing or different types of heuristic rules are most frequently proposed as a solution to this type of problems [2, 3, 7]. Such methods enable a quick achievement of the solution, however, the lack of possibility to obtain the optimal solution is seen as their shared weakness. Moreover, these methods do not allow the assessment of how far the achieved solution lies from the optimal solution.

METHOD FOR DETERMINING THE SEQUENCE OF THE PRODUCTION ORDER WITH THE MINIMAL SUM OF THE DELAY COSTS

This point proposes a method that enables the sequence of the production orders in the two-machine flow system, during which the minimal sum of delay costs is achieved. This method is a generalisation of two methods: a method of scheduling of the production orders with the minimal sum of the delay costs in a one-machine system [2] and a method of scheduling of the production orders with the minimal sum of the delay time in a two-machine system [5].

Each list of orders z_1, z_2, …, z_n requires the following information:
$t_1(i)$ – time for the processing of the order z_i on the machine M_1,
$t_2(i)$ – time for the processing of the order z_i on the machine M_2,
$t_t(i)$ – required deadline for order z_i,
$c(i)$ – the allowed time of the delay of zi order that does not cause charging of the delay costs,
$k(i)$ – unit cost of delay of order z_i, (cost attributable to the unit of time of delay that exceeds value $c(i)$.

In the beginning the sum of the processing times of all undertaken orders on the machine M_1 is determined. This sum was marked as S_1:

$$S_1 = \sum_{i=1}^{n} t_1(i) \tag{1}$$

For each order that can be found on the list, an "indicator of the cost reserve" is determined in such case when said order is realised as the last one. If the order z_i will be executed as the last one on the machine M_1 then the "indicator of the cost reserve" for this order after processing on the machine M_2 will amount to:

$$p(i) = \left(S_1 + t_2(i) - t_t(i) - c(i)\right) \cdot k(i) \tag{2}$$

If $p(i) \geq 0$, then the cost of delay z_i, (connected only to its non-execution) on the basis that it would have been executed as the last one will amount to at least p(i).

The process of finding the optimal solution comprises of two stages. In the first stage of the proposed method the base sequence for which the maximal "indicator of the cost reserve" for the individual order in no bigger than in any other sequence is determined. In the second stage the sequence of the orders that amount to the minimal sum of "indicators of cost reserve" and as a result, a minimum sum of delay costs, is determined.

Remark 1

1. When a parameter $c(i)$ is taken into account it is equal to the introduction of the new deadline $t_t'(i)$ for the realisation of the order z_i:

$$t_t'(i) = t_t(i) + c(i) \tag{3}$$

2. While all $c(i)$ = 0 i $k(i)$ = 1, i = 1, …, n, then the problem comes down to determining the sequence that would give the minimum sum of the delay time, then $p(i)$ means the lack of the time reserve (see [5]). Moreover, when all $t_2(i)$ = 0, the problem is reduced to one-machine problem (see [2]).

Determination of the solutions that give the minimum sum of the delay costs comes in a shape of a tree.

Stage I:

In the beginning, in the first block (the roots of the tree), the $p(i)$ amount for every order is determined in accordance with the formula (2). The order with the lowest $p(i)$ is chosen and placed at the end of the queue (the chosen order is marked as z_j). Next, the time S_1' needed for the execution of the rest of the orders on the machine M_1 (excluding the chosen order z_j) is also determined.

$$S_1' = S_1 - t_1(j) \tag{4}$$

The sum of the "indicators of the cost reserves" (for now only with the consideration the last order in the queue) amounts to .

Afterwards, in block 2 (successor of block 1 in the tree of solutions), the same idea which was used in the roots of the tree is used, however, it is done with the omission of the order z_j. In this case, in order to determine $p(i)$, S_1' is taken into consideration $p(i) = \left(S_1' + t_2(i) - t_t(i) - c(i)\right) \cdot k(i)$.
The order that was chosen in the block 2 (marked as z_k) is put into the queue in the penultimate position. The time needed for the processing of the remaining orders $S_1' = S_1' - t_1(j)$ and the sum of the "indicators of cost reserves" $S_{br} = S_{br}$ + max{p(j);0} are updated.

Subsequently, everything performed in the block 2 is repeated (with the omission of the orders z_j and z_k that were already placed in the last and the penultimate position in the queue) until the block with a number n (leaf in the tree), from which an order is placed to the beginning of the queue, is reached. A base branch of the tree (with the determined sequence of all orders) is also created. The sum of the "indicators of the cost reserves" for the whole branch is marked as S_{brk}.

A sum of the delay costs of all orders S_{kop} after processing on the machine M_2 is determined for the set sequence in the base branch. By all means, if $S_{kop} \geq S_{brk}$. If S_{kop} = 0 then the achieved sequence (not the only option) is optimal on the basis on the sum of delay costs.

Stage II:

If for the sequence determined in the base branch $S_{kop} > 0$, then this sequence does not have to be optimal on the basis of the minimum sum of the delay costs. In this case it is checked from block n-1 to block 1 whether the choice of the next order on the basis of the minimum value $p(i)$ will cause S_{br} to surpass the sum of the delay costs S_{kop} from the base solution. If yes, then the next branch of the tree is not developed.

If not, (which means that S_{br}+max{p(i_k);0} ≤ S_{kop}) then the order z_{ik}, which is placed in the suitable place in the queue, is inserted in the next block (the apodosis of this block) etc.

If another leaf in the tree is obtained (with the determined sequence of all orders), then the sum of "indicators of cost reserves" S_{brk} for this sequence is no higher than the sum of the delay costs S_{kop} in the base solution. In this case the sum of the delay costs for this sequence is determined. If it is lower than S_{kop} from the base solution then S_{kop} is updated and the rest of the branches continues to be checked up to the point where one can still develop some new branches (the S_{kop} will not be exceeded by S_{br}). Finally, the optimal solutions are found within the leaves with the minimal value S_{kop}.

Remark 2.

1. (z_{i1}, z_{i2}, …, , z_{il}, …, z_{in}) shall denote the determined sequence of all orders. If in any block with already determined sequence *n-l* of orders *p(i)* ≤ 0 will be achieved, then the sum of the "indicators of the cost reserves" S_{brk} for the entire branch will be identical to the sum of the "indicators of the cost reserves" S_{br} in this block. Therefore, the sequence of the rest of the orders in the first l positions is, in that case, unrestricted in view of S_{brk}.
2. If for every order the processing time on the machine M_1 is greater than or equal to the processing times of all the other orders on the machine M_2, then the situations when the order is waiting for the processing time on the machine M_2 after being processed on the machine M_1 do not occur. In such case, for each determined sequence, the sum of the delay costs S_{kop} is equal to the sum of the "indicators of the reserve costs" S_{brk}.

EXAMPLES

A couple of examples were presented in order to illustrate the proposed method. In all examples all *c(i)* = 0, *i* = 1, …, *n* were assumed. In the Table 1 the data for the example P1 were presented. The example P1 is combined out of three variants W1, W2 and W3 differing by the unit costs of delay.

In the example P1 the processing times on the machine M_1 are no shorter than the longest processing time on the machine M_2. In this case, in each sequence, any order after being processed on the machine M1 can be immediately processed on the machine M2 (see Fig. 2, 3 and 4). As a consequence, S_{kop} = S_{brk} (see Remark 2.2) occurs in each sequence. In the W1 variant all *k(i)* equal 1, therefore, in this case, the sum of the delay costs will be equal to the sum of the delay times (Remark 1.2).

In Table 2 the achieved results for the example P1 are compiled. The sequences in the base and optimal solution for all three variants were presented. It has to be noted that the breaks in the operation of the machine M_2 do not have any influence on the sum of the delay costs as they do not cause any lengthening of the times of the realisation of the particular orders.

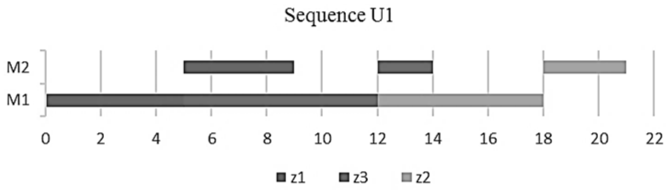

Fig. 2 Gantt chart for the sequence U_1 in the example P1

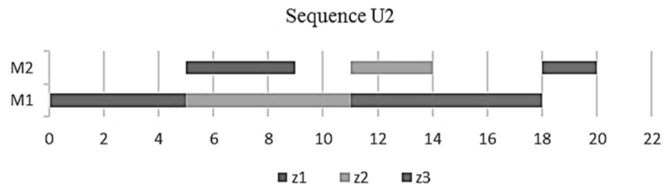

Fig. 3 Gantt chart for the sequence U_2 in the example P1

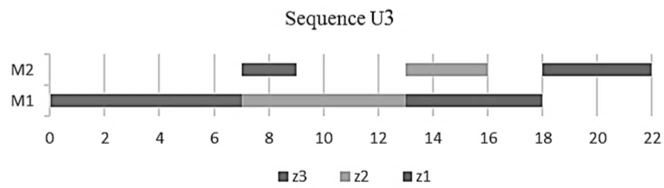

Fig. 4 Gantt chart for the sequence U_3 in the example P1

The tree of solutions for the W1 variant in the example P1 is presented in the Fig. 5 A number of blocks according to the scheduling order (similarly as in the other trees of solution) was presented under (Bi) in the top left corner. The crossed order $\cancel{z_i}$ means that if that order was joined to a given position it would cause a surpassing of the minimum sum of the delay costs, namely, S_{br}+max{p(i);0} ≥ S_{kop}).

The tree of solutions for the variant W2 in the example P1 was presented in the Fig. 6. The base solution turned out to be an optimal one (the same as in case W1).

Figure 7 presents a tree of solutions for the variant W3 in the example P1. The orders in the optimal solution are inverted in relation to the optimal solution for the variants W1 and W2.

Table 1
Data for the example P1

Order	t_1 [min]	t_2 [min]	t_t [min]	Variant W1 k	Variant W2 k	Variant W3 k
z_1	5	4	10	1	1	1
z_2	6	3	14	1	2	2
z_3	7	2	11	1	1	2

Table 2
Compilation of the determined sequences for the example P1

Identification	Sequence	S_{brk}	S_{kop}	Comments
U_1	(z_1, z_3, z_2)	0+3+7 = 10	0+3+7 = 10	base solution for W1
U_2	(z_1, z_2, z_3)	0+0+9 = 9	0+0+9 = 9	optimal solution for W1,
		0+0+9 = 9	0+0+9 = 9	base and optimal solution for W2
U_3	(z_3, z_2, z_1)	0+2·2+12 = 16	0+2·2+12 = 16	base and optimal solution for W3

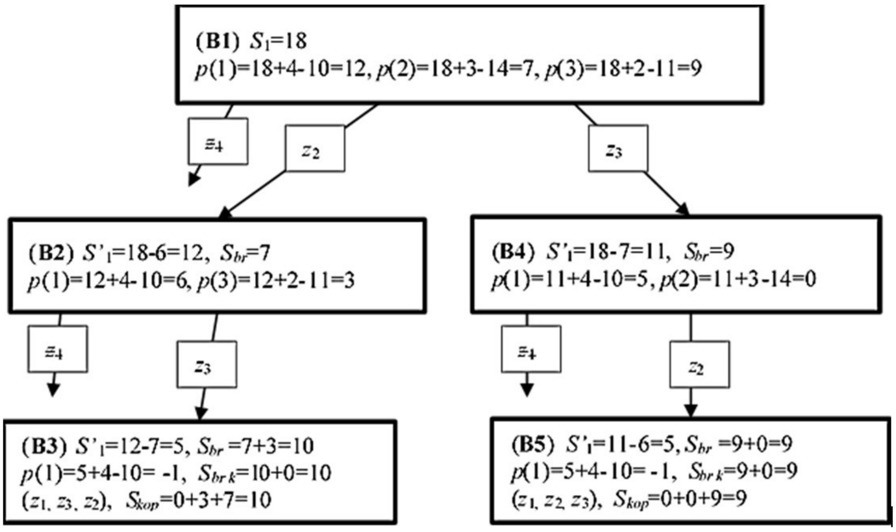

Fig. 5 Tree of solutions for the variant W1 in the example P1

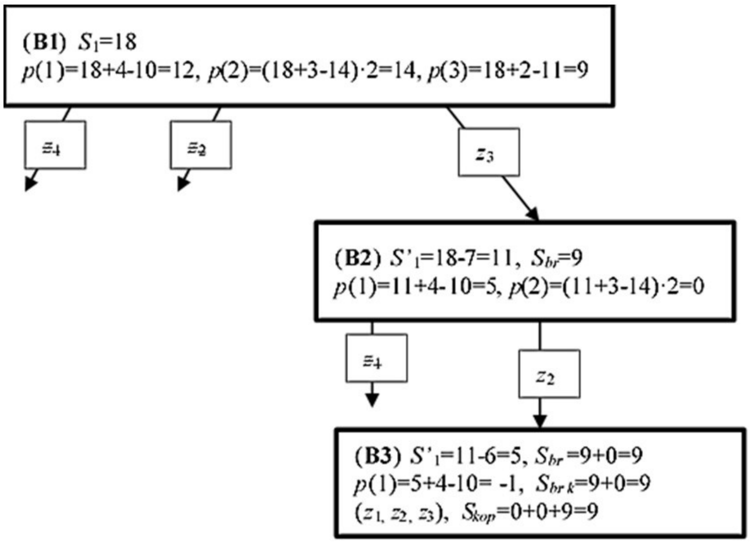

Fig. 6 Tree of solutions for the variant W2 in the example P1

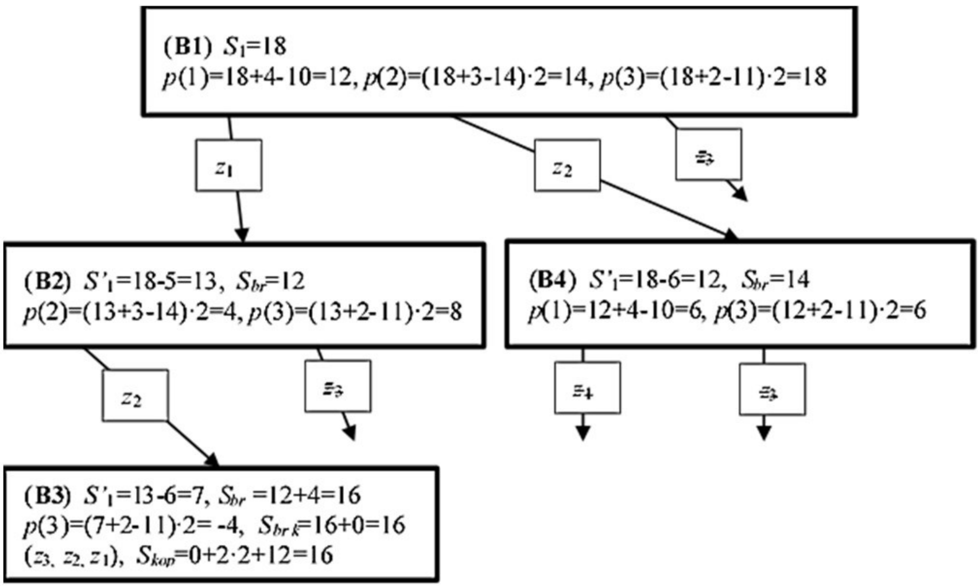

Fig. 7 Tree of solutions for the variant W3 in the example P1

It has to be noted that the optimal solution in relation to the sum of delay costs (in the variant W3) is different from the optimal solution in relation to the sum of the times of delay (variant W1).

In the example P2 (see Table 3) the processing times of the two orders (z_1 and z_3) on the machine M2 are longer than the ones on the machine M1.

As a consequence, a situation in which the order after the termination of the processing time on the machine M1 has to "wait" for the start of the processing on the machine M2 may occur. Similarly to the example P1, in the variant W1, all *k(i)* equal 1, so the sum of the delay costs will be equal to the sum of the delay times (Remark 1.2). In the variant W2 the unit costs of delay of the realisation of the order (z_1, z_3 and z_2) are different.

Gantt charts, determined for the example P2, for the three sequences are presented in the Figures 8, 9 and 10.

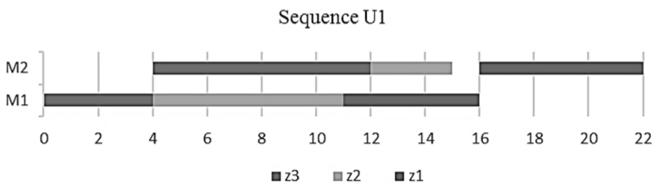

Fig. 8 Gantt chart for the sequence U_3 in the example P1

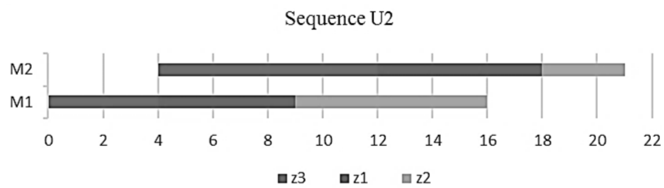

Fig. 9 Gantt chart for the sequence U_2 in the example P2

The obtained results for the example P2 were compiled in the Table 4. The sequences achieved in all leaves of the trees of solutions (U_1 and U_2 for W1 and U_2 and U3 for W2) were presented. It has to be noted that the expectations for the start of the processing on the machine M2 have a significant influence on the sum of the costs of delay.

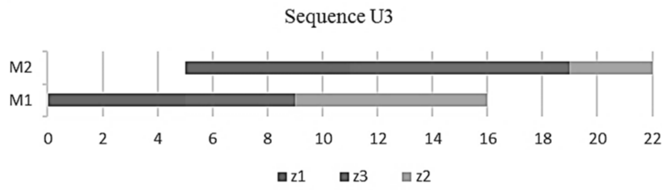

Fig. 10 Gantt chart for the sequence U_3 in the example P2

The sequence U_2 for W1 and U_3 for W2 were determined because of the fact that the sum of the lack of reserves S_{brk} for these sequences did not exceed the sum of delay costs S_{kop} from the base solution, which simultaneously turned out to be an optimal solution (see Fig. 11 and 12). It has to be noted that for the sequence U_1, which is an optimal one in the variant W1 because of the sum of the delay costs, the processing time of all orders amounts to 22 minutes (see Fig. 8), while for the sequence U_2, which is worse because of the sum of the delay costs, this time amounts to 21 minutes. Unfortunately Johnson's algorithm that enables determination of the sequence resulting in the shortest processing time of all orders is useless when it comes to searching for the optimal solution in relation to the minimization of the sum of the delay costs.

The tree of solutions for the variant W1 in the example P2 was presented in the Fig. 11.

Figure 12 presents a tree of solutions for the variation W2 in the example P2. The optimal solution in relation to the sum of the delay costs (in variant W2) is different than the optimal solution in relation to the sum of the delay times (variant W1). It has to be emphasised that the base solution does not have to be in accordance with the increasing deadlines for the realisation of the orders as it was seen in both variants in this example.

In the next example P3 a larger number of orders n = 8 was analysed. For this number of orders the number of all possible sequences amounts to 8! = 40320. Data for this example can be found in Table 5.

The optimal solution for each of the variants was determined with help of a programme created in Matlab, in which the method presented in the point 3 was implemented.

Table 3
Data for the example P2

Order	t_1 [min]	t_2 [min]	t_t [min]	Variant W1 k	Variant W2 k
z_1	5	6	15	1	2
z_2	7	3	11	1	1
z_3	4	8	13	1	2

Table 4
Compilation of the determined sequences for the example P2

Identification	Sequence	Sbrk	Skop	Comments
U_1	(z_3, z_2, z_1)	0+3+7 = 10	0+4+7 = 11	base and optimal solution for W1
U_2	(z_3, z_1, z_2)	0+0+8 = 8	0+3+10 = 13	solution determined for W1,
		0+0+8 = 8	0+3·2+10 = 16	base and optimal solution for W2
U_3	(z_1, z_3, z_2)	0+8+8 = 16	0+6·2+10 = 22	solution determined for W2

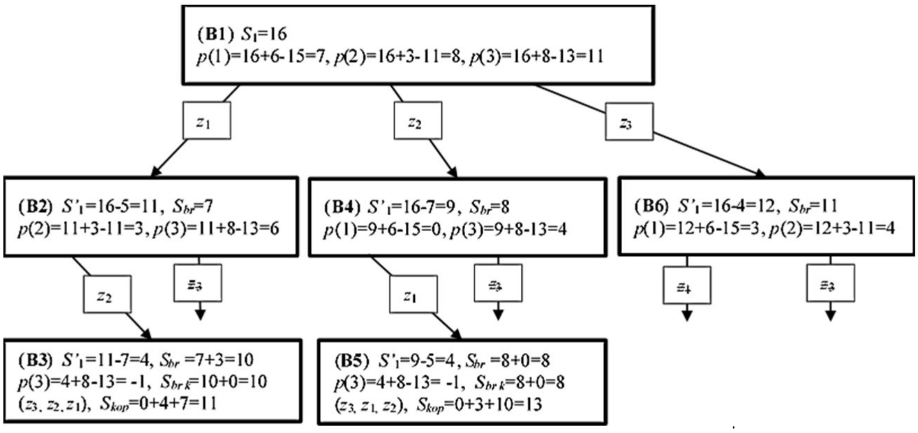

Fig. 11 Tree of solutions for the variant W1 in the example P2

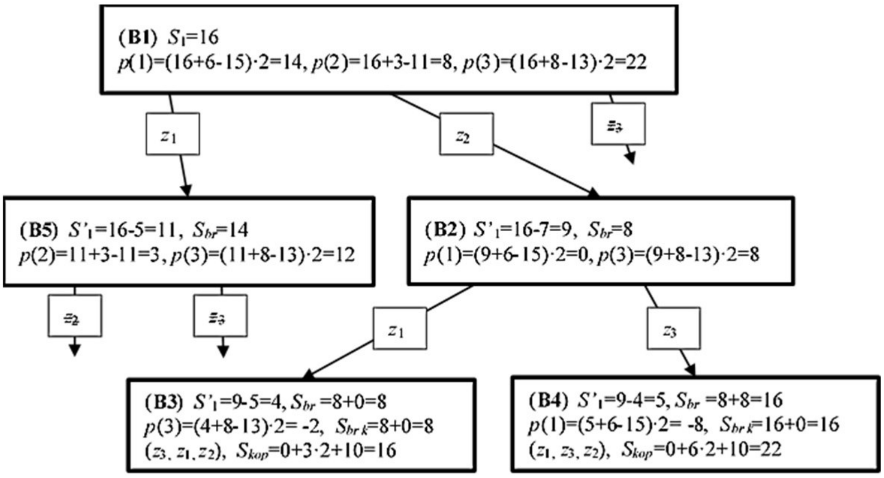

Fig. 12 Tree of solutions for the variant W2 in the example P2

Table 5
Data for the example P3

Order	t_1 [min]	t_2 [min]	t_t [min]	Variant W1 k	Variant W2 k	Variant W3 k	Variant W4 k
z_1	12	1	55	1	1	1	1
z_2	44	5	95	1	1	1	1
z_3	33	3	145	1	2	1	2
z_4	45	6	101	1	1	1	1
z_5	50	7	125	1	2	1	2
z_6	80	4	300	1	1	5	5
z_7	55	8	220	1	1	5	3
z_8	10	2	115	1	1	1	1

Four variants were differing by the values of the unit costs of order delays were defined in the example P3. The base and optimal sequence, the number of determined leaves ad blocks in the tree of solutions and the sum of the delay costs in the base and optimal solution were presented in the Table 6.

Also in this example the base solution, as well as the optimal one, significantly depends on the assumed unit costs of the order delays *k(i)*.

It should be noticed that the search for the optimal solution in relation to the minimum sum of delay costs (variants W2-W4) instead of searching just in relation to the minimum sum of the delay times (variant W1) did not cause significant increase of the number of leaves and blocks required for determining the optimal solutions.

Table 6
Compilation of the results for example P3

Variant	sequence base/optimal	Skop base sol./opt. sol.	number of leaves	number of blocks
W1	(1, 2, 4, 8, 5, 3, 7, 6)	171	2	19
	(1, 2, 4, 8, 3, 5, 7, 6)	154		
W2	(1, 2, 5, 4, 3, 8, 7, 6)	291	4	23
	(1, 2, 8, 5, 3, 4, 7, 6)	183		
	(1, 8, 2, 5, 3, 4, 7, 6)	183		
W3	(1, 2, 4, 8, 5, 7, 3, 6)	341	4	32
	(1, 2, 4, 8, 3, 7, 6, 5)	219		
W4	(1, 2, 5, 4, 3, 8, 7, 6)	497	7	44
	(1, 2, 8, 5, 3, 7, 6, 4)	248		
	(1, 8, 2, 5, 3, 7, 6, 4)	248		

SUMMARY

The problem of determination of the minimal sum of delay costs in processing of the production orders in the two-machine system is much more complicated that a problem with one machine. In contrast to the one-machine system, the size of delay and the delay cost resulting from it for last order can be determined only when the scheduling of all orders is known. It is caused by the possibility of the occurrence of different times of waiting of the order for the start of processing on the machine M_2 after processing on the machine M_1. The only situation when the processing time on the machine M_1 is greater or equal to the processing times of other orders on the machine M_2 there are no occurrences of waiting periods for the start of the processing on the machine M_2 after finishing the processing on the machine M_1. In this case the sum of the delay costs is equal to the sum of the indicators of the reserve costs. In the general case the sum of delay costs may be greater than the sum of the indicators of the reserve costs because of the possible waiting period for the start of processing on the second machine.

It can be stated on the basis of the conducted research that the Johnson's algorithm is useless for solving both the problem of minimization of the sum of the delay costs and the problem of minimization of the sum of the delay times. It may also happen that the sequence determined by the Johnson's algorithm that gives the shortest processing time of all orders in two-machine system generates the highest sum of the delay costs. The sets of orders in which the scheduling is characterised by the lowest sum of delay costs generates the longest processing time of all orders.

The method proposed in this article makes it possible to determine optimal solutions in the relation to the sum of the delay costs in the case of the two-machine system by the use of the so-called indicators of the cost reserves. In the further works the authors plan to generalise the method of determination of the optimal scheduling orders for the n machine systems.

REFERENCES

[1] R. Cupek, A. Ziebinski, L. Huczala and H. Erdogan, "Agent-based manufacturing execution systems for short-series production scheduling", *Computers in Industry*, vol. 82, pp. 245-258, 2016.

[2] R. Dylewski, A. Jardzioch and I. Krebs, "The Optimal Sequence of Production Orders, Taking into Account the Cost of Delays", *Management and Production Engineering Review*, vol. 7, no. 2, pp. 21-28, 2016.

[3] R. Germs, N.D. Van Foreest and O.A. Kilc, "Optimal policies for production-clearing systems under continuous – review", *European Journal of Operational Research*, vol. 255, no. 3, pp. 747-757, 2016.

[4] A. Haned, A. Soukhal, M. Boudhar and N.H. Tuong, "Scheduling on parallel machines with preemption and transportation delays", *Computers & Operations Research*, vol. 39, pp. 374-381, 2012.

[5] A. Jardzioch and R. Dylewski, "Determining the minimum sum of production order delays in a two-machine system", *Management and Production Engineering Review*, vol. 8, No. 3, 2017.

[6] A. Jardzioch, J. Jaskowski. „Calculation of the deadlines for nesting with the possibility of combining orders". *Management Systems in Production Engineering*, no. 1 (17), pp. 22-27, 2015. (doi: 10.12914/MSPE-04-01-2015)

[7] Q. Ji, Y. Wang, and X. Hu, "Optimal production planning for assembly systems with uncertain capacities and random demand", *European Journal of Operational Research*, vol. 253, no. 2, pp. 393-391, 2016.

[8] V.K. Manupatu, G.D. Putnik, M.K. Tiwari, P. Avila and M. Cruz-Cunha, "Integration of process planning and scheduling using mobile-agent based approach in a networked manufacturing environment", *Computers & Industrial Engineering*, vol. 94, pp. 63-73, 2016.

[9] A. Negahban and J.S. Smith, "Simulation for manufacturing system design and operation: Literature review and analysis", *Journal of Manufacturing Systems*, vol. 33, no. 2, pp. 241-261, 2014.

[10] U. Okongwu, M. Lauras, J. Francois and J-Ch. Deschamps, “Impact of the integration of tactical supply chain planning determinants on performance”, *Journal of Manufacturing Systems*, vol. 38, pp. 181-194, 2016.
[11] R.H. Schmitt, M. Ellerich and S. Humphrey, “Multiobjective allocation of customized orders to production-line networks”, *CIRP Annals – Manufacturing Technology*, vol. 65, no. 1, pp. 429-432, 2016.
[12] F. Shrouf, J. Ordieres-Mere, A. Garcia-Sanches and M. Ortega-Mier, “Optimizing the production scheduling of a single machine to minimize total energy consumption costs”, *Journal of Cleaner Production*, vol. 67, pp. 197-207, 2014.

dr Robert Dylewski
University of Zielona Góra, Faculty of Mathematics,
Computer Science and Econometrics
ul. Licealna 9, 65-417 Zielona Góra, POLAND
e-mail: R.Dylewski@wmie.uz.zgora.pl
dr hab. inż. Andrzej Jardzioch, prof. ZUT
West Pomeranian University of Technology Szczecin
Faculty of Mechanical Engineering and Mechatronics
Al. Piastów 19, 70-310 Szczecin, POLAND
e-mail: andrzej.jardzioch@zut.edu.pl
inż. Oliver Dworak
University of Zielona Góra, Faculty of Mathematics,
Computer Science and Econometrics
ul. Licealna 9, 65-417 Zielona Góra, POLAND
e-mail: oli.dwo94@gmail.com