**Management and Production Engineering Review**
Volume 8 • Number 3 • September 2017 • pp. 31–39
DOI: 10.1515/mper-2017-0026

# DETERMINING THE MINIMUM SUM OF PRODUCTION ORDER DELAYS IN A TWO-MACHINE SYSTEM

Andrzej Jardzioch[1], Robert Dylewski[2]

[1] *West Pomeranian University of Technology Szczecin, Faculty of Mechanical Engineering and Mechatronics, Poland*
[2] *University of Zielona Góra, Faculty of Mathematics, Computer Science and Econometrics, Poland*

***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*

Received: 20 December 2016
Accepted: 31 March 2017

Abstract
One of the main tasks in the planning of production processes is to satisfy the needs of the customers in terms of quantity, quality and time. The issue of the timely execution of production orders is becoming increasingly important. Based on the conducted studies it can be concluded that the size of the delay depends on the adopted scheduling of orders. This paper focuses on the problem of implementing a scheduling of production orders that will allow to avoid delays, and in the event such a scheduling is not possible, for minimizing the sum of delays of all the orders. A new algorithm has been proposed that allows to determine of the optimal sequence of production orders with the minimum sum of delays. The considerations have been limited to the issue of a two-machine system in which the orders are carried out in a flow.

Keywords
two-machine flow system, job scheduling, minimum delay times.

## Introduction

The basic paradigm of modern production methods is the desire to fulfill customers' requirements [1, 2]. In the age of universal competition and globalization, enterprises are forced to implement innovative methods for managing and controlling the production process. One of the most innovative production process management concepts is lean management [3]. This concept is based on the rigorous elimination of all forms of waste associated with excessive production, excessive inventory, unnecessary transport and unnecessary time of the orders awaiting processing. One of the ways of eliminating unnecessary downtime is to introduce orders for production in an optimal sequence. The criteria for optimization most often include the maximum lead time of all the production orders, the sum of delays of the orders and the sum of costs associated with the delays [1, 4].

The problem of determining the optimal sequence of production orders has been the subject of many scientific studies [4–7]. The total execution time of all orders is the most common criterion adopted for the determination of the optimal sequence of execution of production orders. This approach allows all orders to be quickly executed but does not consider the required order execution deadlines, and therefore is not able to minimize the delays caused by missed delivery deadlines. In order to achieve success in today's consumer market an enterprise is forced to implement a strategy prioritizing customer satisfaction, and therefore the need to meet the set delivery deadlines.

This article presents a new algorithm that allows the optimal sequence of production orders to be determined in a two-machine system, based on the minimum sum of delays. The developed algorithm utilizes the methodology of branch and bounds, and takes into account the slack time (permissible order delay not leading to delay costs) following processing on the first machine. The proposed algorithm is a certain extension of the algorithm that allows to determine of the optimal scheduling of orders in a single-machine system, presented in detail in paper [1].

The rest of the paper is organized as follows. Section 2 is a detailed presentation of scheduling production orders in a two-machine system. Section 3 describes the algorithm for determining the sequence of production orders with a minimum sum of delays in a two-machine system. In order to provide a better illustration of the proposed methods, examples along with graphical trees of solutions are included in Sec. 4. The paper ends with a summary, which includes the most important conclusions from the carried out works and indicates the issues that the authors plan to address in the course of further research.

## The problem of scheduling production orders processed on two machines

The issue of scheduling production orders has been illustrated using the example of a two-machine manufacturing system implementing unit production and small series production.

The production system consists of an input storage, two machines ($M_1$, $M_2$) and an output storage. The system implements a set of $n$ production orders. It is assumed that all orders are available in the input storage at the start of production ($T_1 = 0$) and may be performed in any sequence. The technological process is carried out in a flow. The execution of each order $z_j$, $j = 1, \ldots, n$ requires the performance of two operations. First operation $O_{1,j}$ is performed on machine $M_1$ and then operation $O_{2,j}$ is carried out on machine $M_2$. Each operation $O_{i,j}$, $i = 1, 2$, $j = 1, \ldots, n$, has an assigned processing time required for its execution – $t_{i,j}$. The required processing time $t_{i,j}$ results from the technological process carried out; it is positive and clearly defined.

Only one operation may be performed on each machine at any given time. The operations are non-preemptive which means that the commenced operation cannot be interrupted. After the completion of operation on machine M2 the order is transferred to the output storage. The output storage capacity allows to store all the executed orders. It was also assumed that the execution of individual orders requires the proper retooling of the machines. The machine retooling time does not depend on the sequence of the orders introduced to production and has been included in the processing time of the individual orders. Furthermore, there are no interruptions in the operation of the machine and in the delivery of orders for production. A diagram of the considered production system is shown in Fig. 1.

Each order has an assigned required completion deadline – $tt_j$.

The problem of scheduling production orders in a two-machine system has been defined as follows:

> It is necessary to use such a sequence of introduction of orders for production U$_i$ (scheduling of orders) that will allow to achieve the minimum sum of delays of all the production orders.

In order to determine the delay of the individual orders the concept of order, delivery date deviation has been introduced. The delivery date deviation was defined as the difference between the actual order execution date – $tr_j$ – and the required completion deadline – $tt_j$ ($\Delta Td_j = tr_j - tt_j$).

If the delivery date deviation of order $z_j$ is greater than zero ($\Delta Td_j > 0$), then the order delay is equal to the order delivery date deviation. If the delivery date deviation is negative or equal to zero ($\Delta Td_j \leq 0$), then the order has been executed before the required deadline. In this case the order execution delay is equal to zero. The purpose of the scheduling process is to find such a scheduling that the sum of delays of all the orders is minimal. The objective function used in the process of determining the optimal scheduling is expressed by formula (1)

$$F_c = \sum_{j=1}^{n} \max\{0, tr_j - tt_j\}, \tag{1}$$

where $F_c$ – the objective function used in the problem of scheduling of production orders, $tr_j$ – the actual execution time of order $z_j$, $tt_j$ – the required completion deadline of order $z_j$.

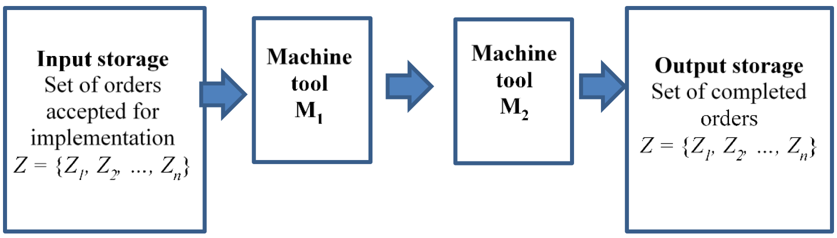


Fig. 1. Diagram of a two-machine production system.

The problem of scheduling production orders defined in such a way is strongly NP-hard. The total number of all possible sequences depends on the number of production orders and equals $n!$. Therefore, finding the optimal solutions with a complete review may not be appropriate in this case, as the computational complexity does not allow to obtain results in an acceptable time (already at 15 production orders the search space is more than $10^{12}$ permutations).

Evolutionary algorithms, swarm algorithms, simulated annealing algorithms or various types of heuristic rules are often suggested for resolving such problems [8]. These methods allow us to obtain solutions quickly, but their common weakness is the inability to obtain the optimal solution. They also do not allow us to estimate how far the obtained solution is from the optimal solution. Because of that investigations are being carried out to find a method that would allow to identify the optimal sequence of orders for a given objective function [8–11].

Johnson's algorithm is often used in order to solve two-machine problems. It allows us to find the shortest paths between every pair of vertices in graphs and can be used to find a scheduling with which the total duration of all orders will be minimal (the $F2||C_{\max}$ problem). Unfortunately, this algorithm cannot be used to find the scheduling which allows to minimize of the sum of delays of the production orders. In the defined two-machine flow system, the production does not run smoothly (uninterrupted). In a situation where following the processing on machine M1 an order should be forwarded to processing on machine $M_2$, but machine $M_2$ is busy, the order is blocked on machine $M_1$ until machine $M_2$ is free. In this situation the 'wait-time' of the order on machine $M_1$ until the completion of processing of another order on machine $M_2$ extends the execution of individual production orders. The frequency of occurrence of 'wait-time' of the production orders on machine M1 depends on the adopted scheduling of the production orders and impacts their execution time.

## Method for determining the sequence of production orders with a minimum sum of delays

In this point a method is proposed that allows to determine the sequence of production orders enabling us to minimize the sum of delays. This method is a certain modification of the method for the single-machine problem, presented in paper [1] and it consists of two stages. In the first stage the base sequence is determined, i.e. a sequence for which the maximum lack of slack time for a single order is not greater than in all the other sequences. In the second stage the sequences of orders providing the minimum sum of delays are determined.

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

- $t_1(i)$ – the processing time for order $z_i$ on machine $M_1$,
- $t_2(i)$ – the processing time for order $z_i$ on machine $M_2$,
- $t_t(i)$ – the required deadline for order $z_i$.

The determination of the base sequence begins with the determination of the sum of the processing times of all orders received on machine $M_1$. This sum is denoted by $S_1$:

$$S_1 = \sum_{i=1}^{n} t_1(i). \tag{2}$$

For each order included on the list, the lack of slack time is determined in the event it is performed as the last one. If order $z_i$ is executed as the last one on machine $M_1$, then the lack of slack time for this order after processing on machine $M_2$ will be:

$$p(i) = S_1 + t_2(i) - t_t(i). \tag{3}$$

If $p(i) \geq 0$, then the delay in the execution of order $z_i$ will amount to at least $p(i)$, if it is carried out as the last one.

The determination of solutions with a minimum sum of delays has the form of a tree.

Stage I:

At the beginning, in the first block (the root of the tree), the value $p(i)$ is determined for each order according to formula (2). Any of the orders for which $p(i)$ is the smallest is selected and placed at the end of the queue (the selected order is marked as $z_j$). Then the time that needed to carry out the rest of the orders on machine $M_1$ is determined (excluding the selected order $z_j$); this equals

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

The sum of the lack of slack time (for the time being, only taking into account the last order in the queue) is $S_{br} = \max\{p(j); 0\}$.

Then, in block 2 (the successor of block 1 in the tree of solutions), what has been done in the root of the tree is performed, but excluding the order that has already been put at the end of the queue. However, in order to determine $p(i)$, $S_1'$ ($p(i) = S_1' + t_2(i) - t_t(i)$) it now taken into account. The order selected in block 2 is designated as $z_k$ and is inserted into the queue as the second to last. The time required for processing the remaining orders is updated $S_1' := S_1' - t_1(k)$ and the sum of the lack of slack time $S_{br} := S_{br} + \max\{p(k); 0\}$.

Then what has been done in block 2 is repeated (ignoring the orders already inserted as the last and second to last in the queue) until block of $n$ is reached (leaf in the tree), from which the order is inserted to the front of the queue. In this way the base branch in the tree is obtained (with the established sequence of all orders). The sum of the lack of slack time for the entire branch is designated as $S_{brk}$.

For the established sequence in the base branch we determine the sum of the delays of all orders $S_{op}$ following processing on machine $M_2$. Of course $S_{op} \geq S_{brk}$. If $S_{op} = 0$, then the received sequence is optimal due to the sum of delays.

Stage II:

If in a sequence determined in the base branch the sum $S_{op} > 0$, then this sequence will not necessarily give the minimum sum of delays. We then check from block $n-1$ to block 1, whether the selection of the next order, in terms of the minimum value $p(i)$, would cause the $S_{br}$ to exceed the sum of delays $S_{op}$ of the base solution. If that is the case, another branch is not expanded. If not (i.e, $S_{br} + \max\{p(i_k); 0\} \leq S_{op}$), the selected order $z_{ik}$ is entered into the next block in the appropriate place in the queue, etc.

If the next leaf in the tree is reached (with an established sequence of all orders), then the sum of the lack of slack time $S_{brk}$ for this sequence is no higher than the sum of delays $S_{op}$ for the base solution. Therefore, the sum of delays for this sequence is determined. If it is lower than $S_{op}$ from the base solutions, then the $S_{op}$ is updated and checking the subsequent branches is continued until they can be developed (until $S_{op}$ is not be exceeded by $S_{br}$). Finally, the optimal solutions are in the leaves with the minimum $S_{op}$ value.

Caution:

The established sequence of all orders is denoted as $(z_{i1}, z_{i2}, \ldots, z_{ik}, \ldots, z_{in})$. If some block with an already established sequence of $n-k$ orders shows $p(i) \leq 0$ for all the other orders, then the sum of the lack of slack time $S_{brk}$ for the entire branch will be the same as the sum of the lack of slack time $S_{brk}$ in that block. Therefore, the sequence of the remaining orders on the first $k$ positions is not important due to $S_{brk}$.

## Examples

Four examples have been presented in order to better illustrate the proposed method. Data for Example 1 are shown in Table 1.

It should be noted that the shortest time of processing orders on machine $M_1$ is not shorter than the longest time of processing on $M_2$. In such case, for every sequence any order whose processing on machine $M_1$ has been completed can be immediately processed on machine $M_2$ (see Figs. 2 and 3). As a consequence for every sequence $S_{op} = S_{brk}$.

Table 1
Data for Example 1.

| Order | Processing time on $M_1$ $t_1$ [min] | Processing time on $M_2$ $t_2$ [min] | Required deadline $t_t$ [min] |
|---|---|---|---|
| $z_1$ | 5 | 4 | 10 |
| $z_2$ | 6 | 3 | 14 |
| $z_3$ | 7 | 2 | 11 |

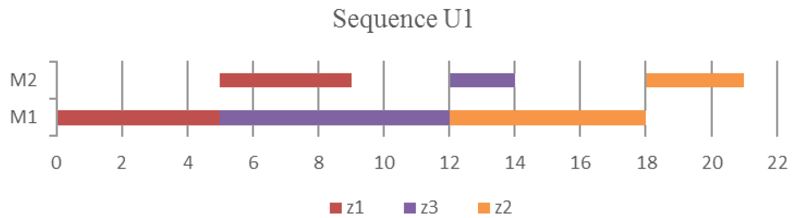


Fig. 2. Gantt chart for sequence $U_1$ in Example 1.

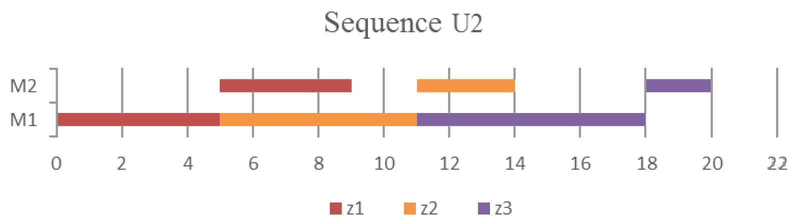


Fig. 3. Gantt chart for sequence $U_2$ in Example 1.

The obtained results are summarized in Table 2. Sequences for the base solution ($U_1$) and optimal solution ($U_2$) were given. It should be noted that the interruptions in work on machine $M_2$ do not have any effect on the sum of the delay times.

Figure 4 shows the tree of solutions for Example 1. In the upper left corner, the block number according to the designation sequence is given under (Bi). The crossed out order $z_i$ indicates that the inclusion of this order to the given item would cause the minimum sum of delays to be exceeded, i.e. $S_{br} + \max\{p(i); 0\} \geq S_{op}$.

In Example 2 parameters $t_2$ and $t_t$ are changed for orders $z_3$ in relation to Example 1 (see Table 3). In this example the shortest time of processing orders on machine $M_1$ is also not shorter than the longest time of processing on $M_2$.

The obtained results are summarized in Table 4. The sequence established in the base solution is also the optimal solution. It should be noted that the sequence in the base solution is not in compliance with the increasing completion deadlines.

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

| Identification | Sequence | $S_{brk}$ | $S_{op}$ | Comments |
|---|---|---|---|---|
| $U_1$ (blok B3) | ($z_1$, $z_3$, $z_2$) | $0+3+7=10$ | $0+3+7=10$ | base solution |
| $U_2$ (blok B5) | ($z_1$, $z_2$, $z_3$) | $0+0+9=9$ | $0+0+9=9$ | optimal solution |

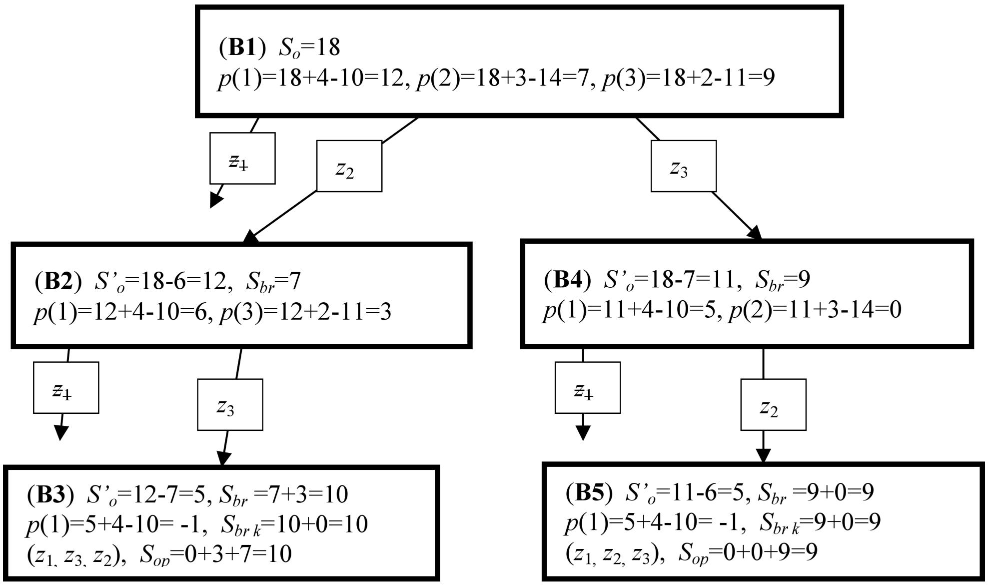

Fig. 4. Tree of solutions for Example 1.

Table 3
Data for Example 2.

| Order | Processing time on $M_1$ $t_1$ [min] | Processing time on $M_2$ $t_2$ [min] | Required deadline $t_t$ [min] |
|---|---|---|---|
| $z_1$ | 5 | 4 | 10 |
| $z_2$ | 6 | 3 | 14 |
| $z_3$ | 7 | 1 | 13 |

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

| Identification | Sequence | $S_{brk}$ | $S_{op}$ | Comments |
|---|---|---|---|---|
| $U_1$ (blok B3) | ($z_1$, $z_2$, $z_3$) | $0+0+6=6$ | $0+0+6=6$ | base and optimal solution |

Figure 5 shows the tree of solutions for Example 2.

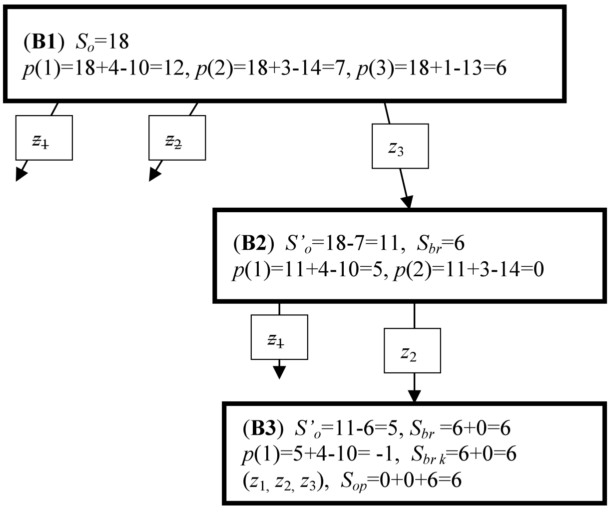


Fig. 5. Tree of solutions for Example 2.

In Example 3 (see Table 5) the times of processing on machine $M_2$ are longer than on $M_1$ for some orders. For this reason, there are situations in which orders whose processing on machine $M_1$ has been completed have to "wait" for the start of processing on machine $M_2$.

Table 6 shows the results obtained for this example. Sequences obtained in all four leaves of the tree of solutions are presented. The sequences $U_2$, $U_3$ and $U_4$ have been determined because the sum of the lack of slack time $S_{brk}$ for these sequences did not exceed the sum of delays $S_{op}$ from the base solution, which also turned out to be optimal. It should be noted that for sequence $U_1$, which is optimal in terms of the sum of delays, the processing time of all orders is 18 minutes (see Fig. 6), and for sequence $U_4$, which is much worse in terms of the sum of delays, this time is 15 minutes. This is the shortest time of processing of all the orders (see Fig. 7). Moreover, the sum of wait time for the start of processing on machine $M_2$ for sequence $U_4$ amounted to $1 + 2 = 3$ minutes, and for the optimal sequence $U_1$ it was much longer: $1 + 6 = 7$ minutes. As can be seen the Johnson's algorithm, which allows us to designate the sequence providing the shortest time of processing of all the orders (in this example – 15 minutes) is not useful in terms of the minimization of the sum of the delays.

Figure 8 shows the tree of solutions for Example 3.

Finally Example 4 was presented with four orders (see Table 7). In this example there are orders which have a longer processing time on machine $M_1$ and an order which has a longer processing time on machine $M_2$.

Table 5
Data for Example 3.

| Order | Processing time on $M_1$ $t_1$ [min] | Processing time on $M_2$ $t_2$ [min] | Required deadline $t_t$ [min] |
|---|---|---|---|
| $z_1$ | 1 | 5 | 18 |
| $z_2$ | 4 | 3 | 8 |
| $z_3$ | 2 | 6 | 12 |

Table 6
Summary of the designated sequences for Example 3.

| Identification | Sequence | $S_{brk}$ | $S_{op}$ | Comments |
|---|---|---|---|---|
| $U_1$ (blok B3) | $(z_2, z_3, z_1)$ | $0+0+0=0$ | $0+1+0=1$ | base and optimal solution |
| $U_2$ (blok B4) | $(z_3, z_2, z_1)$ | $0+1+0=1$ | $0+3+0=3$ | |
| $U_3$ (blok B6) | $(z_2, z_1, z_3)$ | $0+0+1=1$ | $0+0+6=6$ | |
| $U_4$ (blok B7) | $(z_1, z_2, z_3)$ | $0+0+1=1$ | $0+1+3=4$ | |

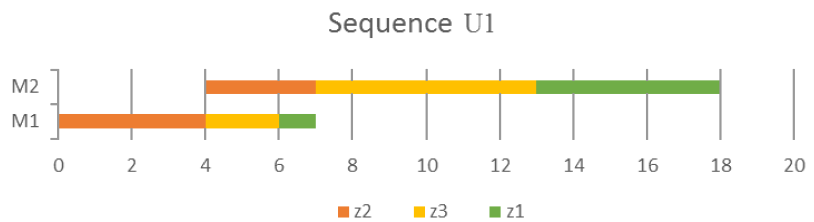


Fig. 6. Gantt chart for sequence $U_1$ in Example 3.

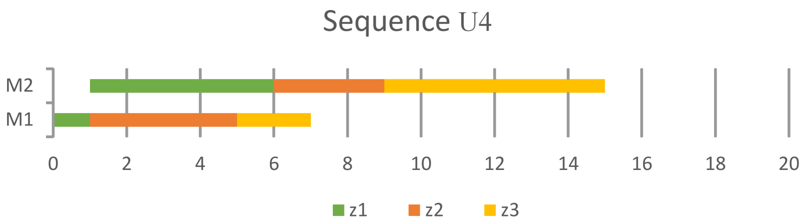

Fig. 7. Gantt chart for sequence $U_4$ in Example 3.

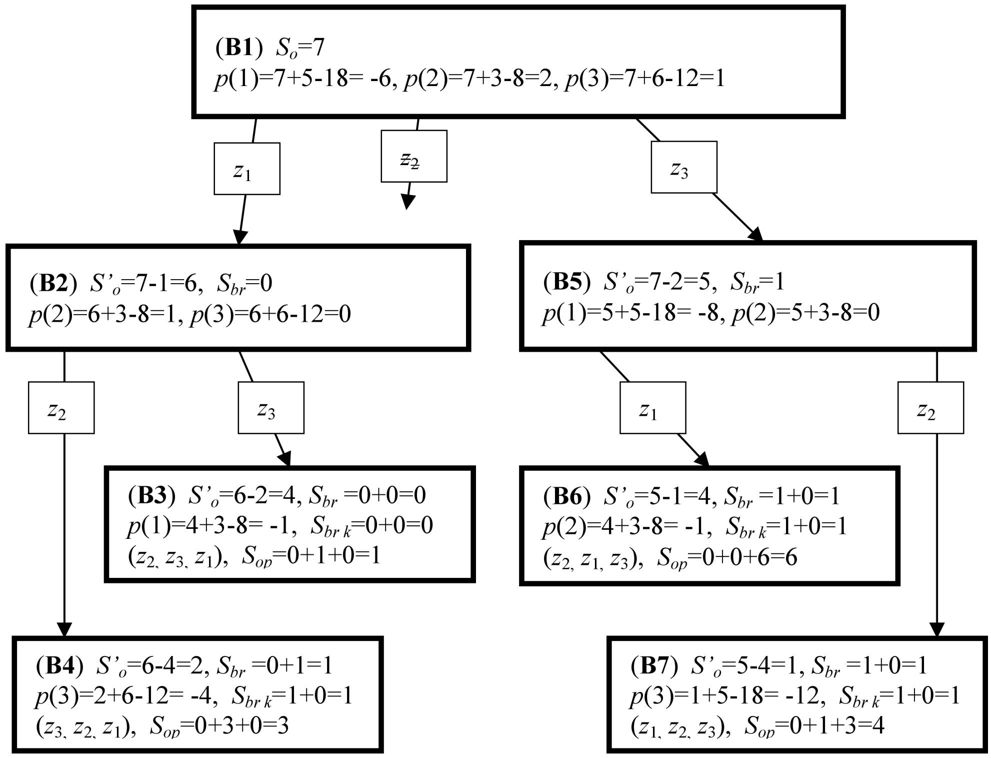

Fig. 8. Tree of solutions for Example 3.

Table 7
Data for Example 4.

| Order | Processing time on $M_1$ $t_1$ [min] | Processing time on $M_2$ $t_2$ [min] | Required deadline $t_t$ [min] |
|---|---|---|---|
| $z_1$ | 10 | 2 | 70 |
| $z_2$ | 5 | 11 | 20 |
| $z_3$ | 20 | 1 | 60 |
| $z_4$ | 15 | 6 | 50 |

Table 8 shows the results obtained for this example. Sequences obtained in all three leaves of the tree of solutions are presented. All three sequences turned out to be optimal. It should be noted that for solution $U_1$ the time of processing of all the orders is 52 minutes (see Fig. 9), and there are no cases of orders waiting for the start of processing on machine $M_2$ after the completion of processing on machine $M_1$. Meanwhile for solution $U_3$ the time of processing of all the orders is 51 minutes (see Fig. 10) and there is a wait time (1 minute) for order $z_1$ (because $t_1(1) = 10$ and $t_2(2) = 11$). Despite that, both sequences give the minimum sum of delays.

Figure 11 shows the tree of solutions for Example 4.

Table 8
Summary of the determined sequences for Example 4.

| Identification | Sequence | $S_{brk}$ | $S_{op}$ | Comments |
|---|---|---|---|---|
| $U_1$ (blok B4) | $(z_2, z_4, z_3, z_1)$ | $0+0+0+0=0$ | $0+0+0+0=0$ | base and optimal solution |
| $U_2$ (blok B7) | $(z_2, z_4, z_1, z_3)$ | $0+0+0+0=0$ | $0+0+0+0=0$ | optimal solution |
| $U_3$ (blok B9) | $(z_2, z_1, z_4, z_3)$ | $0+0+0+0=0$ | $0+0+0+0=0$ | optimal solution |

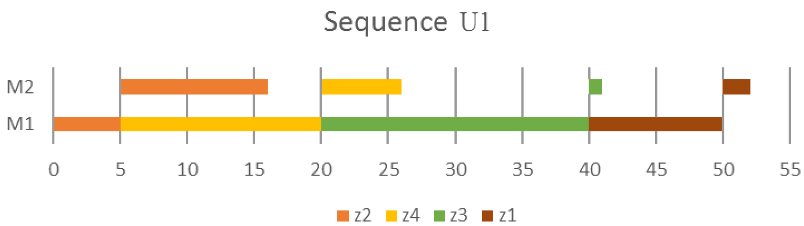


Fig. 9. Gantt chart for sequence $U_1$ in Example 4.

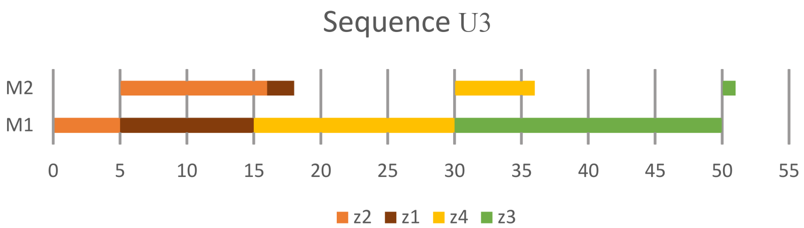


Fig. 10. Gantt chart for sequence $U_3$ in Example 4.

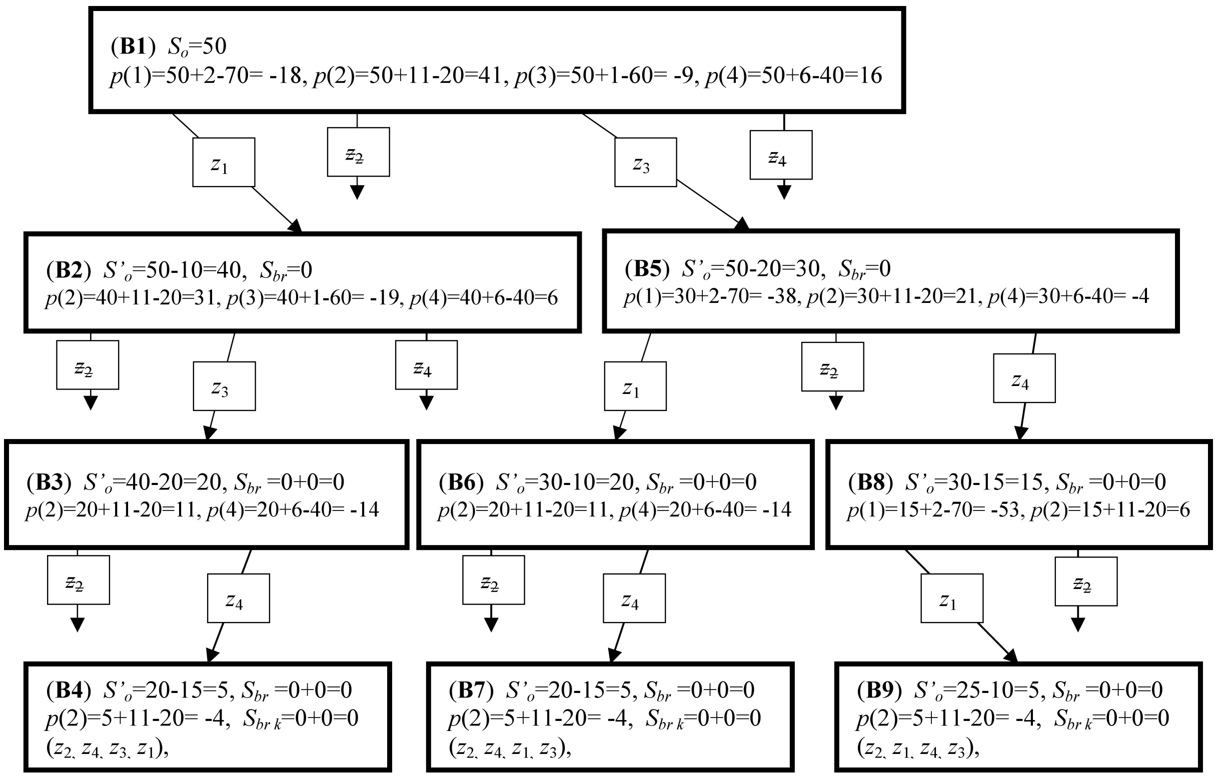


Fig. 11. Tree of solutions for Example 4.

## Summary

The problem of determining the minimum sum of delays in the processing production orders in a two-machine system is much more complex than in the case of a single-machine system. In contrast to single-machine systems, the time of delay of the last order can be determined only once the scheduling of all the orders is known. This is due to the occurrence of cases where orders await the start of machining on machine $M_2$ following the completion of processing on machine $M_1$. If for every order the time of processing on machine $M_1$ is greater or equal to the time of processing of the remaining orders on machine $M_2$, then there are no cases of orders awaiting the start of machining on machine $M_2$ following the completion of processing on machine $M_1$. In such case the sum of the delay times is equal to the sum of the lacks of slack time. However, generally the sum of the delay times can be greater than the sum of the lack of slack time, precisely because of the possible waiting-time for the start of machining on the second machine.

Unfortunately Johnson's algorithm is quite useless for the problem of minimization of the sum of delays. It may happen that the scheduling providing the shortest processing time of all the orders in a two-machine system will give the greatest sum of delays. In addition, the scheduling giving the longest processing time of all orders gives the smallest sum of delays.

The algorithm proposed in this paper allows us to find the optimal solutions from the point of view of the sum of delay times in a two-machine system, by utilizing the so-called lack of slack time ratio.

Further work is required in which the authors are planning to extend the proposed method to the problem of minimization of the sum of delay costs associated with the execution of all the orders.

## References

[1] Dylewski R., Jardzioch A. Krebs I., *The Optimal Sequence of Production Orders, Taking into Account the Cost of Delays*, Management and Production Engineering Review, 7, 2, 21–28, 2016.

[2] Okongwu U., Lauras M., Francois J., Deschamps J-Ch., *Impact of the integration of tactical supply chain planning determinants on performance*, Journal of Manufacturing Systems, 38, 181–194, 2016.

[3] Cherrafi A., Elfezazi S., Chiarini A., Mokhlis A. Benhida K., *The integration of lean manufacturing, Six Sigma and sustainability: A literature review and future research directions for developing a specific model*, Journal of Cleaner Production, 139, 828–846, 2016.

[4] Aljorephani S. K., ElMaraghy H. A., *Impact of Product Platform and Market Demand on Manufacturing System Performance and Production Cost*, Procedia CIRP, 52, 74–79, 2016.

[5] Yang D-L., Hou Y-T., Kuo W.-H., *A note on a single-machine lot scheduling problem with indivisible orders*, Computers and Operations Research, 79, 34–38, 2017.

[6] Schmitt R. H., Ellerich M., Humphrey S., *Multi-objective allocation of customized orders to production-line networks*, CIRP Annals – Manufacturing Technology, 65, 1, 429–432, 2016.

[7] Hoogeveen H., Lente C., Tkindt V., *Rescheduling for new orders on a single machine with setup times*, European Journal of Operational Research, 223, 1, 40–46, 2012.

[8] Jardzioch A. Bulwan K., *The prioritisation of production orders under the bee colony algorithm*, Advances in Manufacturing Science and Technology, 37, 4, 49–59, 2013.

[9] Ji Q., Wang Y., Hu X., *Optimal production planning for assembly systems with uncertain capacities and random demand*, European Journal of Operational Research, 253, 2, 393–391, 2016.

[10] Germs R., Van Foreest N.D., Kilc O.A., *Optimal polices for production-clearing systems under continuous – review*, European Journal of Operational Research, 255, 3, 747–757, 2016.

[11] Wan H., Wang Q., *Asymptotically-optimal component allocation for Assemble-to-Order production – inventory systems*, Operations Research Letters, 43, 3, 304–310, 2015.