Int j simul model 15 (2016) 1, 29-41
ISSN 1726-4529 Original scientific paper

HYBRID ALGORITHM BASED ON PRIORITY RULES FOR SIMULATION OF WORKSHOP PRODUCTION

Zupan, H.; Herakovic, N.; Starbek, M. & Kusar, J.
University of Ljubljana, Faculty of Mechanical Engineering, Askerceva 4, 1000 Ljubljana, Slovenia
E-Mail: hugo.zupan@fs.uni-lj.si, janez.kusar@fs.uni-lj.si

Abstract
The proposed hybrid algorithm is a combination of heuristic algorithm extended with priority rules, discrete event simulation and genetics algorithm. It takes into account 11 different priority rules and scenarios, is based on the assumption that for a realistic workshop scheduling of orders, it is necessary to consider real throughput times of the operations, otherwise the obtained scheduling of orders is not suitable for the industrial environment. The simulation result for the proposed model is the optimal sequence of selected orders for the selected time interval while taking into account three criteria: the minimum flow time of all orders, the maximum average utilization of workplaces, and the minimum waiting time of the orders. Because the mentioned criteria are usually mutually exclusive the advantage of the proposed model is that we can find the optimum with respect to all three criteria. In the paper, an example of the application of the proposed model is shown.

(Received in March 2015, accepted in July 2015. This paper was with the authors 1 month for 1 revision.)

Key Words: Workshop Scheduling, Discrete Event Simulation, Priority Rules, Optimization of Workshop

1. INTRUDUCTION

In a workshop type of production, workplaces and working tools are arranged according to the processing operations, i.e. all the lathes are in the lathe-shop, the orders or work pieces are transported from workplace to workplace, either individually or in small batches (see Fig. 1).

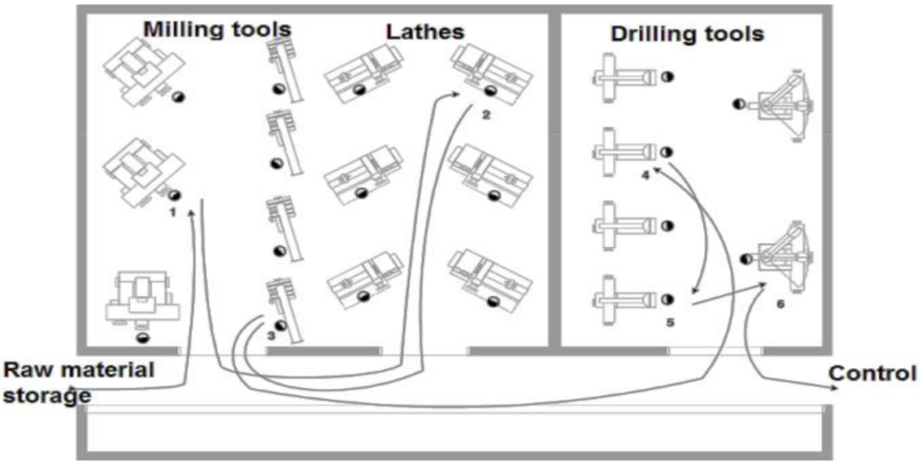


Figure 1: Example of the principle workflow in a workshop.

Compared to other production types, a workshop type of production has the great advantage of being very flexible, so it can adapt to different orders and to a different sequence of the orders. The disadvantage of a workshop type of production is the messy material flow of the orders and thus the long flow times and higher cost of the orders [1].

At the stage of the planning and controlling of the orders (individual work pieces or a small batch of pieces) that came to the workshop during the selected time interval, we have to distinguish between [2] (see Fig. 2):

DOI:10.2507/IJSIMM15(1)3.319

- the throughput time of each operation *i* of order *j*, and
- the flow time of order *j*, which covers all throughput times of all the operations in the order.

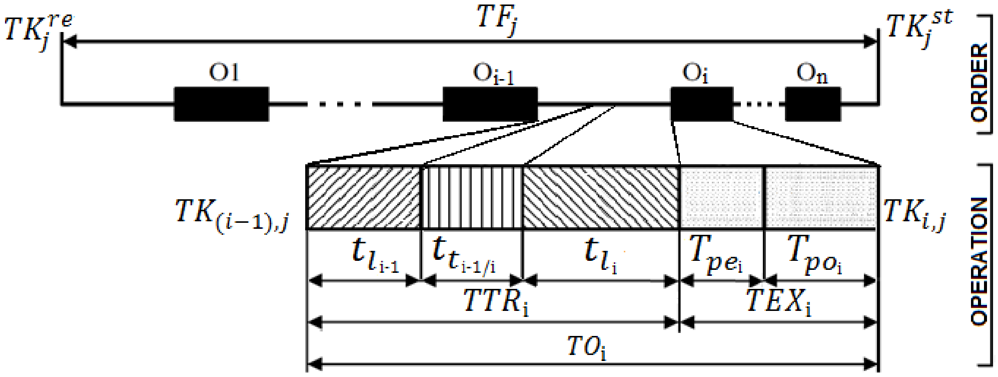


Figure 2: Structure of the throughput time of operation *i* and the flow time of order *j*.

TF_j – Flow time of order *j* (*j* = 1, 2, …, *m*)
TO_i – Throughput time of operation *Oi* (*i* = 1, 2, …, *n*)
$t_{l_{i-1}}$ – Time of lying after operation O*i*-1
$t_{t_{i-1/i}}$ – Time of transportation from operation O*i*-1 to operation *Oi*
t_{l_i} – Time of lying before operation *Oi*
T_{pe_i} – Set-up time of operation *Oi*
T_{po_i} – Processing time of operation *Oi*
TTR_i – Transition time of operation *Oi*
TEX_i – Time of execution of operation *Oi*

The flow time of order *j* represents the time interval calculated from the time point of the removal of the raw material for the order from storage to the time point of the storing of the product of the order:

$$TF_j = TK_j^{st} - TK_j^{re} \tag{1}$$

TF_j – Flow time of order *j* [Wd]
TK_j^{st} – Time point of storing the finished order *j*
TK_j^{re} – Time point of removal of the order *j* slug from storage

The throughput time of operation *i* is the shortest unit of the order flow time, and by adding up the throughput times of the operations, the flow time of the order is determined. It therefore applies that the flow time of order *j* is equal to the sum of the throughput times of *i* operations that must be performed on order *j*.

$$TF_j = \sum_{i=1}^{n} TO_{i,j} = \sum_{i=1}^{n} (TK_{i,j} - TK_{(i-1),j}) \tag{2}$$

TF_j – Flow time of order *j* [Wd]
$TO_{i,j}$ – Throughput time of operation *i* of order *j* [Wd]
$TK_{i,j}$ – Time point of the finished operation *i* of order *j*
$TK_{(i-1),j}$ – Time point of the finished operation (*i*-1) of order *j*
i – The number of the operation (*i* = 1, 2, …, *n*)
j – The number of the order (*j* = 1, 2, …, *m*)

Therefore, if we want to determine the flow time of order *j*, we must first determine the throughput times of all *i* operations of order *j*, and by adding together the throughput times of the planned operations, we get the flow time of the order, TF_j.

In order to interpret and determine the throughput time of the operation the funnel model is used [3], which represents a workplace or machine on which the particular operation is performed. During the selected time interval (usually at least one month), any workplace receives an arbitrary number of orders, and for each order we must measure:

- RECEPTION OF THE ORDERS:
  - the number of the order which the workplace RECEIVED,
  - the time when the order was RECEIVED at the workplace, and
  - the content of the work to be done for the order at the workplace [Eh].
- DEPARTURE OF THE ORDERS:
  - the number of the order which DEPARTED from the workplace,
  - the time when the order DEPARTED from the workplace, and
  - the content of the work done for the order at the workplace [Eh].

The results of the measurements of the flow of orders through the workplace are entered into the throughput diagram or the load–effect diagram [3] from which we can read the throughput times of the operation of all the orders that left the observed workplace.

When all the weighted throughput times of all *k* orders (*k* = 1, 2, ..., *z*) which departed workplace *i* during the selected time interval are known [3], we can calculate the weighted mean throughput time of workplace *i* (operation *i*):

$$\overline{TO}_i^* = \frac{\sum_{k=1}^{z} TO_{ik}^* \times TORD_{ik}}{\sum_{k=1}^{z} TORD_{ik}} = \overline{TTR}_i + \overline{TEX}_i \tag{3}$$

The content of the work at workplace *i* for order *k* is:

$$TORD_{ik} = Tpe_{ik} + M_k \times t_{e1_k} \tag{4}$$

$\overline{TO}_i^*$ – Weighted mean throughput time of workplace *i* [Wd]
TO_{ik}^* – Weighted throughput time of workplace *i* for order *k* [Wd]
$TORD_{ik}$ – The content of the work at workplace *i* for order *k* [min/order]
$\overline{TTR}_i$ – Weighted mean transition time of the orders at workplace *i* [Wd]
$\overline{TEX}_i$ – Weighted mean execution time of the orders at workplace *i* [Wd]
Tpe_{ik} – Set-up time of workplace *i* for order *k* [Wd]
M_k – Number of work pieces of order *k* [piece/order]
t_{e1_k} – Time per unit of order *k* [min/piece]

To obtain the real weighted mean throughput times of the orders, $\overline{TO}_i^*$, or the real weighted mean transition times of the orders, $\overline{TTR}_i$, in a company with a workshop organized production we have to record the load–effect diagram for the selected time interval for each workplace in the workshop. These diagrams allow us to calculate the achieved real weighted mean throughput times and the real weighted mean transition times of the orders that have been fulfilled at a workplace. The resulting matrix of the weighted average throughput times $\|\overline{TO}_i^*\|$ and the weighted mean transition times $\|\overline{TTR}_i\|$ will serve us as the basis for realistic planning of the orders for the next time interval (next month).

For heuristic planning of the sequence of the execution of orders at the workplaces or work tools, we use priority rules. Priority rules determine the order in which the orders that are queuing up at a workplace will be fulfilled. Especially in an individual or small-scale production – a workshop organized production –, many orders with a different number of operations, different times of execution and different products in small batches pass through the workplaces.

For workplace scheduling, elementary or combined priority rules can be used [4].

Elementary priority rules are:

1. FIFO (First in First out): the orders are lined-up according to their reception.
2. LIFO (Last in First out): the order that arrives last is first in line.
3. SPT (Shortest Processing Time): the waiting order with the shortest processing time at a workplace gets the highest priority.
4. LPT (Longest Processing Time): the waiting order with the longest processing time at a workplace gets the highest priority.
5. SLT (Shortest Processing Time Left): the order with the shortest processing times of all unrealized operations gets the highest priority.
6. LLT (Longest Processing Time Left): the order with the longest processing times of all unrealized operations gets the highest priority.
7. STT (Shortest Total Time): the order with the shortest total time of all operations gets the highest priority.
8. LTT (Longest Total Time): the order with the longest total time of all operations gets the highest priority.
9. LOR (Least Operations Remaining): the order with the lowest number of still unrealized operations gets the highest priority.
10. MOR (Most Operations Remaining): the order with the highest number of still unrealized operations gets the highest priority.
11. ST (Slack Time): the preferred time operation of the part with the lowest time reserve.

In addition to the elementary priority rules, there are also combined priority rules, which are several elementary rules joined to form combined priority rules, such as:

- ST/SPT (Slack Time against Shortest Processing Time): the total priority represents the ratio of ST to SPT.

Workshop scheduling is a demanding combinatorial optimization problem. Literature review [4, 5] shows that the following methods are used for solving the scheduling of orders in a workshop:

- Exact methods which are typically used to determine the schedule of orders only in selected cases and are practically useless in a real environment. These include mixed integer programming, branch and bound etc. [6].
- The methods of a travelling salesman where all the points of a given graph are visited exactly once and we want the total distance to be the shortest path [7].
- Heuristic approaches based on priority rules [8].
- The methods of transfer bottlenecks where we locate the bottleneck in the existing production and try to overcome it so that we get more efficient workplaces [9].

With the above mentioned methods, only easier cases are successfully resolved; for solving complex problems, other methods were developed:

- Discrete event simulation, which today is becoming an increasingly useful tool for solving the scheduling of orders. Its major advantage is that it does not consume material, energy and resources, it only requires data. Thus, different variants of production are tested in advance as we search for an optimal solution [10, 11].
- Tabu-search (TS), which is one of the most commonly used metaheuristics in combinatorial optimization problems. TS algorithm uniquely uses search history to escape from a local optimum, as well as for the execution of search strategies. A simple algorithm TS is used as a basic local search with the best improvement. In addition, short-term memory is also used to escape from a local optimum and to avoid repetition [12, 13].
- Genetic algorithms, which can be described as a computational model of the evolutionary processes. In each iteration (generation), a certain number of operators is applied to the current population of the specimens so that the next generation of specimens is created. Commonly used are the crossover operator, which combines two or more specimens to gain a new specimen, and the mutation, which causes the self-adaptation of specimens [14, 15].

- Simulated annealing, which is analogous to the process of the cooling of the material. The basic idea is to allow the steps that return a worse than the current solution, which allow us an escape from the local optimum. The probability of such steps is decreasing during the search [16].
- Swarm intelligence is a field of artificial intelligence which uses the crowd or particle swarm to solve complex problems. The inspiration for many algorithms which fall in this area can be found in nature. Among the most popular are the Firefly algorithm [17], particle swarm optimization [17], Cuckoo search [18], the artificial bee colony algorithm [19, 20], ant colony optimization [17], and others.
- Neural networks are in most cases an adaptive system that changes its structure during learning. The structure varies based on the external and internal information provided by a network of processes. Neural networks are used for modelling complex relationships or for data mining; i.e. the search for a previously unknown function which we cannot determine analytically [21, 22].

In this article, we are going to schedule the orders in a workshop using a hybrid algorithm which is a combination of the heuristic algorithm with priority rules, discrete event simulation and genetic algorithms.

In the simulation, we will use 11 different priority rules, and with the help of the selected genetic algorithm we will determine which priority rule gives the optimal sequence of orders in relation to the flow time of the selected orders, the average utilization of machines and the waiting time of the orders.

For the simulation, we will use the throughput times of operations, which has not yet been used in literature [23].

2. WORKSHOP SIMULATION

It is characteristic of a workshop that in the selected time period (week or month) several different orders are received with a different number of operations, different set-up times Tpe_{ik} and different processing times of operations T_{po_i}.

Orders consist of one or more operations to be carried out in different workplaces. Planning the sequence of orders in the workshop production represents a complex problem. In the absence of an exact scheduling method, heuristics, which is based on the priority rules, is used in practice to solve this problem. When planning the sequence, we may follow different goals. Typical goals are:

- the minimization of order flow time [24],
- the maximization of utilization capacity [24],
- the minimization of order delay deadlines, and
- the minimization of the manufacturing cost, cost of material flow or storage cost [24, 25].

Conflicts often occur between the given goals [3].

2.2 Building the simulation model of a workshop production

For the purpose of building the model and implementing the simulation, we use the software package Tecnomatix Plant Simulation. The program is a leading software solution for modelling and simulation and it is based on the theory of discrete events and is object-oriented. The anticipated advantages of using this program are [26]:

- as much as 6 percent savings on the initial investment,
- an increase in existing system productivity by as much as 20 percent,
- a reduction of the new system costs by as much as 20 percent,
- an optimization of resource consumption and their re-use, and

- a reduction of stock and flow time by as much as 60 percent.

The objective of the simulation is to determine the optimal sequence of the execution of orders to be carried out in the selected time frame. With the help of simulations, we will examine in what sequence the orders should be fulfilled in order to obtain the shortest flow time of all the orders or the optimal utilization of tools or the shortest waiting time of the orders.

The simulation model of a workshop production will be built so that it will allow the execution of *j* orders (*j* = 1, 2, ..., *m*) at *i* workplaces (*i* = 1, 2, ..., *n*). The known data for each order are:

- the set-up time of workplaces ($T_{pe_{i,j}}$),
- time per unit (t_{e1_k}),
- the number of work pieces per order ($M_{i,j}$), and
- the weighted mean transition time of the workplaces ($\overline{TPR_{i,j}}$).

Raw material will be stored in the material storage (S_m) and completed orders in the products storage (S_p).

The purpose of the research will be to obtain data about the execution time of the orders in workshop production using a simulation model.

The whole workshop system will be entered into a virtual factory or a simulation model based on the actual production system (see Fig. 3).

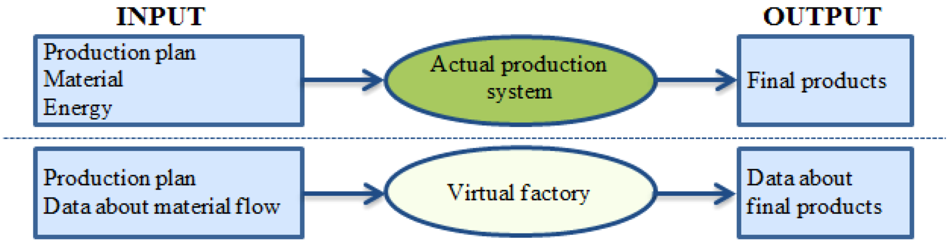


Figure 3: The basic principle of the virtual factory [27].

When building the model of an observational workshop production, the model is designed in such a way that it will be able to use the input data of an actual production. The model collects all the essential characteristics of the production system. The data obtained with the help of the virtual factory are used to help detect the waiting times in buffers and to eliminate them. On the basis of the suppositions of the virtual production process and the characteristics of the actual production process, we designed a logical scheme of a workshop production (see Fig. 4).

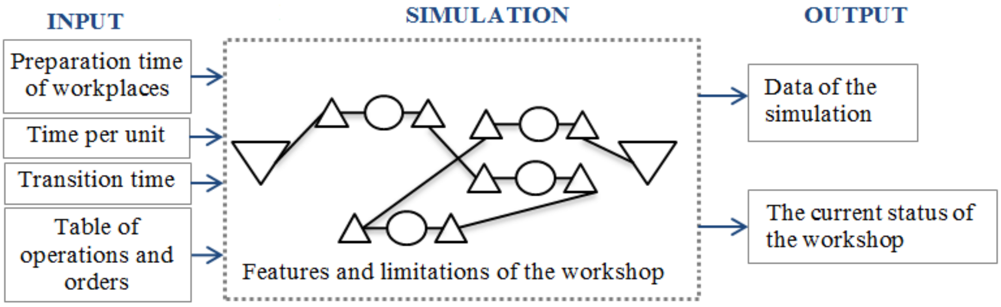

INPUT
SIMULATION
OUTPUT
Preparation time of workplaces
Time per unit
Transition time
Table of operations and orders
Features and limitations of the workshop
Data of the simulation
The current status of the workshop

Figure 4: The logical scheme of the model of a workshop production.

On the basis of the data obtained, we designed a simulation model. The simulation illustrates the current state of the workshop, covering all its properties in the selected time frame. Each workplace represents a single machine, which includes a buffer of needs, a place buffer and the space where the operation is performed. The simulation output data will include:

- a table of the flow time of all orders, using the optimal sequence of the orders,
- a Gantt Chart of scheduling scenarios for the orders, and
- histograms of the utilization of machines and the waiting of the orders to be fulfilled.

2.2 Simulation model of a workshop

The simulation model of a workshop was built in the Plant Simulation software package. The simple logical dependences of the production processes in the model are denoted by standard software package objects, and the complex logical dependences are denoted with methods or libraries in the programming language SimTalk [28-30]. The program consists of 72 standard objects of the software package and more than 300 lines of logical dependences written in SimTalk.

The basic data for the construction of a virtual factory and the creation of a simulation model is the number of workplaces where operations are being performed:

▽ – material storage (S_m) and products storage (S_p),

○ – technological operations O_i (i = 1, ..., n), and

△ – buffer of needs BN_i and place buffer BP_i (i = 1, ..., n).

The model is designed so that an object that has the name "Warehouse Material" represents the storage of raw material (S_m) from which orders go to the workshop. From the material storage (S_m), an initial order N_k is transported to the first operation $O1$. Between the observed operation of stocking-out the material (S_m) and the next operation ($O1$), there must elapse transition time $\overline{TTR}$, which is determined on the basis of the computer-recorded load–effect diagram for each workplace. After the transportation, the prescribed first operation ($O1$) in the chosen order according to the technological process is performed. The execution time of the operation consists of the set-up time of the workplace (T_{pe}) and the processing time ($M_k \times te1$).

If it so happens that there are several orders at the same operation, then the orders are awaiting a free operation in the buffer of needs located in front of the operation. When order N_k is fulfilled at operation $O1$, it is transported to the next operation $O2$. This process is repeated many times until the last operation of the order or the entire production process is finished, so that order N_k has arrived in the warehouse of products (S_p).

When all scheduled orders N_k (k = 1, 2, …, z) at workplaces WP_i (i = 1, 2, …, n) are fulfilled, we get the optimal flow time of the execution of orders k.

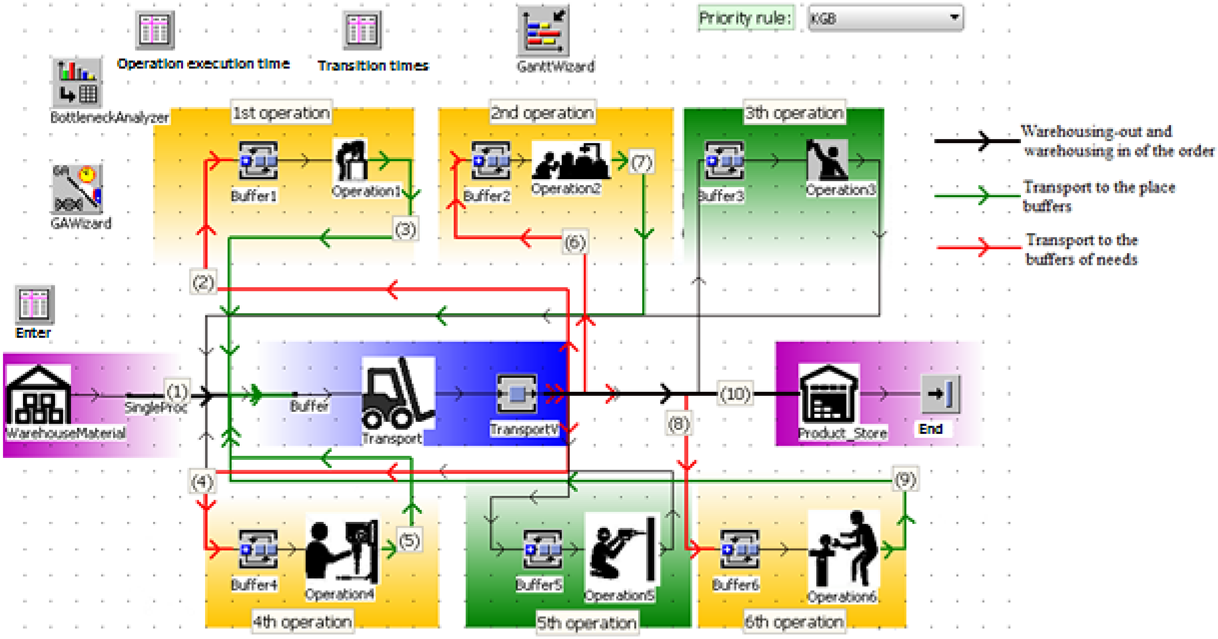
Logical dependences between objects are linked by code that is written in the SimTalk language. The code is primarily used for the allocation of orders between workplaces and for the determination of various priority rules.

With the code for tracking the sequence of operations of the orders, the simulation model correctly allocates orders to the next required position. With the help of the code, it sends the order to the buffer of needs of the next workplace of the order. The code which allocates the order to the proper position or operation is:

```
if @.OperationIn /= " Product Store" then
        @.move(”Buffer”+ @.OperationIn);
elseif @. OperationIn = "Product Store" then
        @.move(Final_Store);
end;
```

An important piece of information for carrying out the simulation flow of the orders in a workshop is the manufacturing processes of executing the orders (technology procedure and sequence of operations).

In Fig. 5, there are shown the simulation model of a workshop which covers the workplaces that perform the operations of order N_k and the transport routes as defined by the course of the manufacturing process of order N_k.

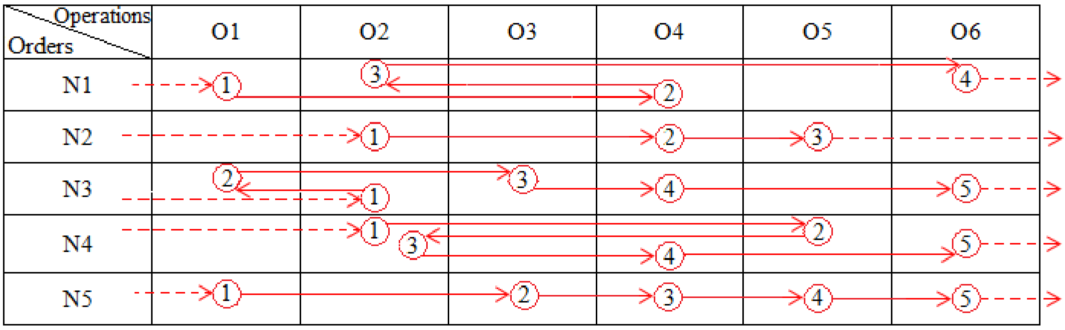


Priority rule: KGB

Figure 5: The simulation model of a workshop and the flow of order N_k.

3. TESTING OF THE SIMULATION MODEL OF THE WORKSHOP

The plan for the workshop is to produce 5 orders with up to 6 operations/necessary tools. For each order, the technology procedure and the sequence of operations are known (see Fig. 6):



Operations / Orders	O1	O2	O3	O4	O5	O6
N1	1	3		2		4
N2		1		2	3	
N3	2	1	3	4		5
N4		1 3		4	2	5
N5	1		2	3	4	5

Figure 6: The matrix of the type and sequence of operations.

The necessary data for running the simulation of the orders are:

1. The matrix of the set-up times of the workplaces $\|Tpe^{*}_{k,j}\|$ [min/order]:

$$\|Tpe\| = \begin{array}{cccccccl} O1 & O2 & O3 & O4 & O5 & O6 & & \\ \multicolumn{6}{c}{\xrightarrow{\text{Operations of order}}} & & \\ 20 & 10 & 30 & 25 & 60 & 18 & N1 & \\ 30 & 15 & 25 & 30 & 70 & 20 & N2 & \\ 25 & 5 & 30 & 25 & 80 & 15 & N3 & \text{Orders} \\ 40 & 20 & 23 & 12 & 65 & 23 & N4 & \\ 5 & 15 & 18 & 10 & 30 & 42 & N5 & \downarrow \end{array}$$

2. The matrix of times per unit $t_{e1_{k,j}}$ [min/piece]:

$$\xrightarrow{\text{Operations of order}}$$

$$\|t_{e1}\| = \begin{array}{c|cccccc|cl} & \text{O1} & \text{O2} & \text{O3} & \text{O4} & \text{O5} & \text{O6} & & \\ & 25 & 30 & 40 & 20 & 40 & 5 & \text{N1} & \\ & 45 & 25 & 30 & 60 & 38 & 30 & \text{N2} & \\ & 16 & 60 & 15 & 33 & 8 & 13 & \text{N3} & \text{Orders} \\ & 7 & 14 & 50 & 3 & 23 & 13 & \text{N4} & \\ & 50 & 3 & 19 & 15 & 62 & 52 & \text{N5} \downarrow & \end{array}$$

3. The vector of lot sizes of the orders $M_{k,j}$ [piece/order]:

$$\|M\| = \begin{array}{|c|cl} 200 & \text{N1} & \\ 400 & \text{N2} & \\ 50 & \text{N3} & \text{Orders} \\ 150 & \text{N4} & \\ 20 & \text{N5} \downarrow & \end{array}$$

4. The matrix of the weighted mean transition times of the orders $\overline{TTR}_{k,j}$ [Wd]:

$\|\overline{TTR}\|$ =

To From	O1	O2	O3	O4	O5	O6
O1	■	1	4	1	3	1
O2	1	■	2	2	3	2
O3	4	2	■	3	3	1
O4	1	2	3	■	3	3
O5	3	3	3	3	■	1
O6	1	2	1	3	1	■

The purpose of the simulation is to determine how different start scenarios affect the execution of all orders. In other words, we are checking different scenarios of the production plan, and the simulation for each scenario shows the desired results.

The number of different scenarios depends on the number of orders that we allocate. This increases with the factorial of the orders.

In this particular production plan, there are 5 orders which means that the number of all possible scenarios of scheduling is 120. The simulation model has determined the optimal initial sequence (N5, N2, N4, N1, N3) and the optimal flow time of the execution which is 42 Wd 11 h 40 min.

In practice, it is unusual to optimize the flow time of the execution of orders independently of the priority rules, therefore, priority rules are taken into account for the scheduling of orders.

Each priority rule has a different code in the simulation model. Below is a code example for priority rule SPT (the shortest execution time of the operation *TEX* has the highest priority). The code calculates the execution time of the operation for each order at a specific operation/workplace, and then compares all the orders in the buffer of needs in front of each operation, and the order with the shortest execution time of the operation goes first into execution.

The code is:

```
for j:=1 to Operation1.Setup_time.YDim loop
        if Operation1.Setup_time[1,j]=@.Name then
                @.Operation_time:=Operation1.Setup_time[2,j]+Operation1.Processing_time[2,j];
```

The simulation was performed for different priority rules with the purpose of determining how different priority rules and start scenarios affect the execution of orders. Some rules allow the execution of different start scenarios (FIFO, LIFO, LOR, MOR and ST), other rules allow just one start scenario (SPT, LPT, SLT, LLT, STT and LTT). If the priority rule allows different scenarios, we determine the best scenario with the help of genetic algorithm [31]. The obtained results are given in Table I.

Table I: The obtained results according to different priority rules.

Priority rule	Optimal sequence of orders	Flow time [Wd]	Average utilization of all workplaces [%]	Wait time of orders [Wd]
FIFO	N5, N2, N4, N1, N3	42 Wd 11 h 40 min	23,59 %	82 Wd 15 h 32 min
LIFO	N4, N5, N1, N3, N2	42 Wd 17 h 15 min	23,46 %	84 Wd 3 h 5 min
LOR	N2, N1, N4, N3, N5	44 Wd 4 h 57 min	22,67 %	105 Wd 14 h 7 min
MOR	N4, N5, N3, N1, N2	44 Wd 19 h 20 min	22,37 %	89 Wd 14 h 42 min
ST	N5, N4, N3, N2, N1	42 Wd 17 h 15 min	23,46 %	155 Wd 9 h 7 min
SPT	N4, N3, N2, N5, N1	44 Wd 19 h 20 min	22,37 %	90 Wd 9 h 20 min
LPT	N2, N3, N4, N1, N5	45 Wd 4 h 57 min	22,67 %	84 Wd 3 h 5 min
SLT	N5, N3, N4, N1, N2	45 Wd 18 h 5 min	21,91 %	74 Wd 1 h 52 min
LLT	N2, N1, N4, N3, N5	44 Wd 4 h 57 min	22,67 %	106 Wd 27 min
STT	N5, N3, N4, N1, N2	45 Wd 18 h 5 min	21,91 %	80 Wd 20 h 56 min
LTT	N2, N1, N4, N3, N5	44 Wd 4 h 57 min	22,67 %	105 Wd 14 h 7 min

The histogram of the flow times of all the orders for all priority rules which were considered in the simulation is shown in Fig. 7.

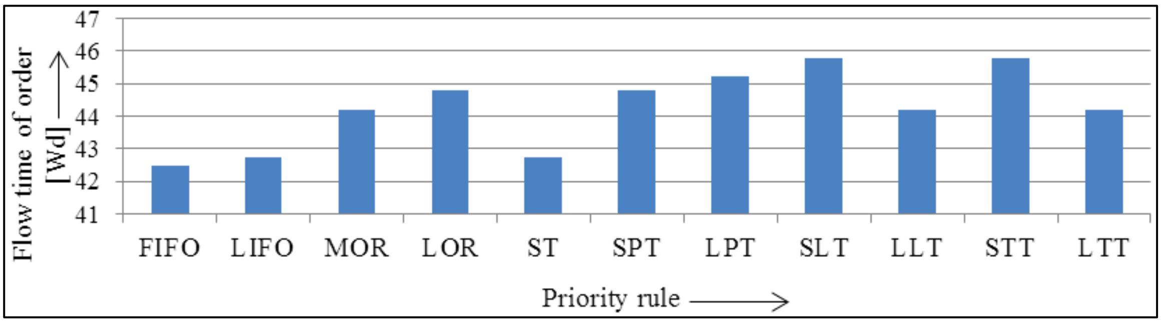


Figure 7: Histogram of flow times in relation to the use of different priority rules.

The histogram of the average utilization of all tools and the histogram of the waiting times of the orders for all priority rules which were considered in the simulation are shown in Fig. 8.

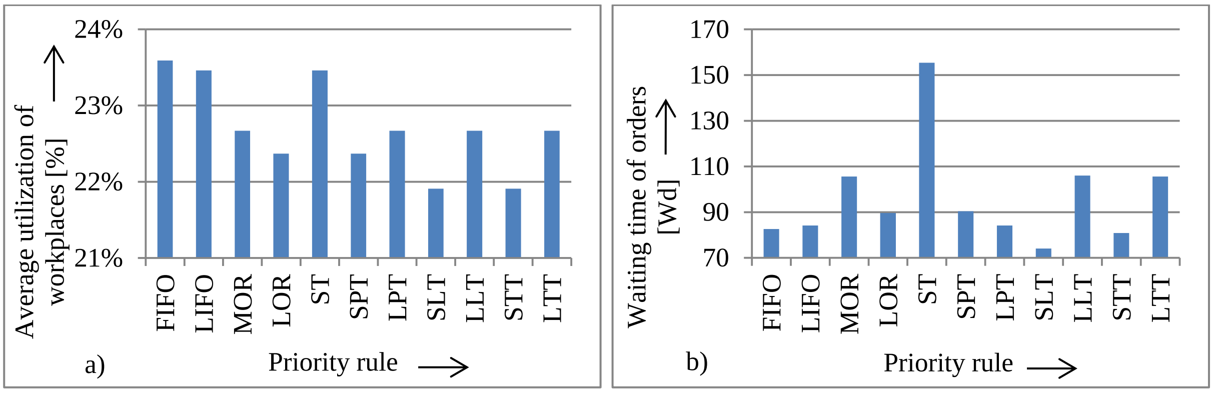
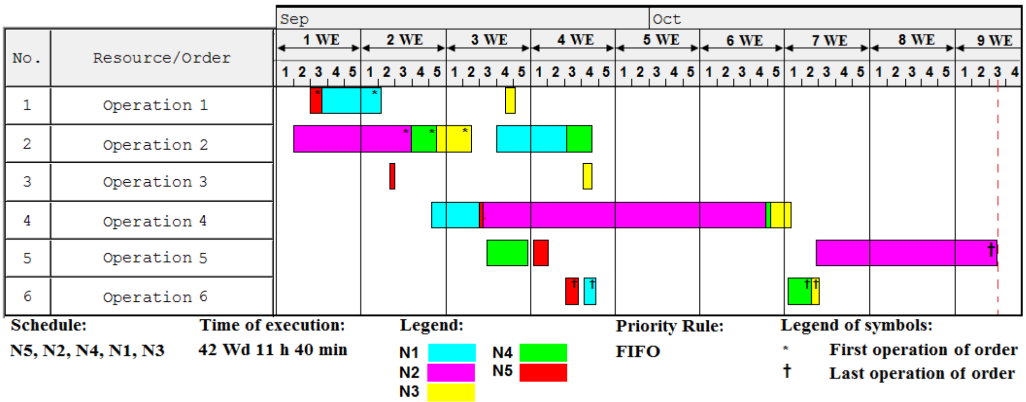


Figure 8: Histogram: a) of the average utilization of all tools in relation to the use of different priority rules, b) of the waiting time of all the orders in relation to the use of different priority rules.

From the histogram in Fig. 7, we can establish that the shortest flow time of the orders is with the use of the FIFO priority rule. Fig. 9 shows the Gantt chart for the best scenario (FIFO rule).

Gantt chart in Fig. 10 shows the only possible scenario for the priority rule SPT.

The simulation model allows a quick simulation of scheduling different scenarios of the orders at workplaces. Since in practice there is usually the need to schedule a larger number of orders with a different number of operations, we examined how an increase of the number of orders and an increase of the number of operations affect the execution time of the simulation. The results of the test are shown in Fig. 11.



Schedule: N5, N2, N4, N1, N3

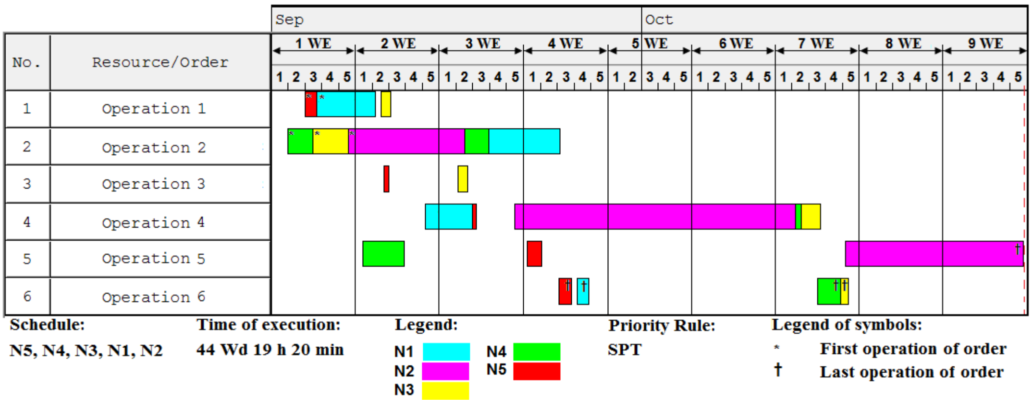
Time of execution: 42 Wd 11 h 40 min

Legend: N1 🟦 N2 🟪 N3 🟨 N4 🟩 N5 🟥

Priority Rule: FIFO

Legend of symbols:
* First operation of order
† Last operation of order

Figure 9: Gantt chart of the best scenario for the FIFO priority rule.



Schedule: N5, N4, N3, N1, N2

Time of execution: 44 Wd 19 h 20 min

Legend: N1 🟦 N2 🟪 N3 🟨 N4 🟩 N5 🟥

Priority Rule: SPT

Legend of symbols:
* First operation of order
† Last operation of order

Figure 10: Gantt chart for the scenario using the SPT priority rule.

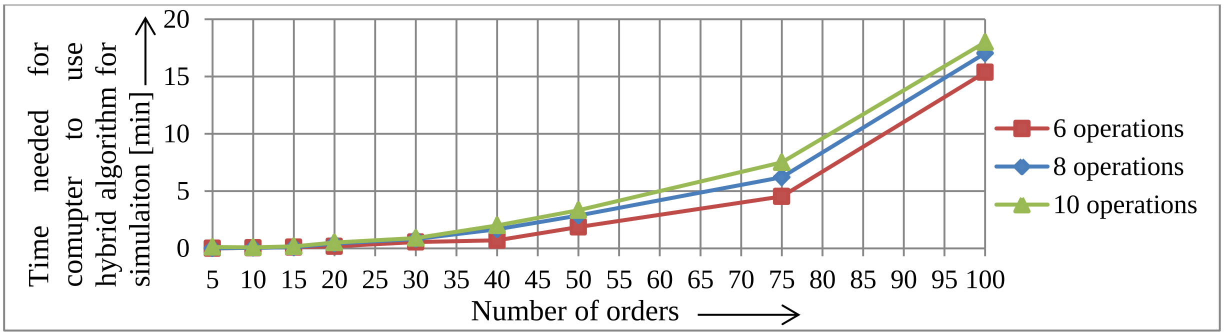


Figure 11: Simulation time depending on the number of orders and the number of operations.

In Fig. 11, it can be seen that the time for implementing different scenarios of the simulated scheduling of orders with a computer (Intel Core i5-2500 CPU, 3.30 GHz, Win 7) lengthens relatively slowly, indicating that the proposed simulation model is suitable for solving concrete examples from industrial practice.

4. CONCLUSION

In this paper, we used a simulation model in Plant Simulation to simulate the flow of the material of orders in workshop production.

We have presented the theoretical basis on which the workshop production operates and shown guidelines for obtaining real-time data (such as transition time between workplaces, preparation time of workplaces and time per unit) to carry out the simulation. The simulation was performed with the use of the throughput times of the operations of the orders, which produces realistic results. In the model, we have seen how different scheduling of orders and different priority rules affect the total flow time, the waiting time of all orders, and the average utilization of workplaces. Another advantage of the simulation model of a workshop is that we can quickly check (see Fig. 11 above) whether we will be able to carry out the intended production plan or when we will be able to carry out a certain number of orders.

In the model, we can quickly and easily change the process parameters and promptly see what implications these parameters have for the production process. This is particularly important due to the fact that a workshop production is very flexible, and by using the what-if scenarios in advance we can remove all unnecessary costs.

A future research challenge will be to discover how to simulate the transition from a workshop production into a cellular production, and how to optimize the cellular production using the proposed hybrid algorithm.

ACKNOWLEDGEMENT

The research has been co-financed by the Operational Programme for Human Resources Development (KROP 13), Priority Axis: Promoting entrepreneurship and adaptability, Main type of Activity 1.1.: Experts and researchers for competitive enterprises.

REFERENCES

[1] Ficko, M.; Palcic, I. (2013). Designing a layout using the modified triangle method, and genetic algorithms, *International Journal of Simulation Modelling*, Vol. 12, No. 4, 237-251, doi:10.2507/IJSIMM12(4)3.244

[2] Wiendahl, H. P. (2008). *Betriebsorganisation für Ingenieure*, Hanser, München, 262-269

[3] Nyhuis, P.; Wiendahl, H.-P. (2012). *Logistische Kennlinien*, 3rd ed., Springer-Verlag, Berlin

[4] Pinedo, M. L. (2012). *Scheduling: Theory, Algorithms, and Systems*, Springer Science+Business Media, LLC, New York

[5] Goncalves, J. F.; Mendes, J. J. M.; Resende, M. G. C. (2005). A hybrid genetic algorithm for the job shop scheduling problem, *European Journal of Operational Research*, Vol. 167, No. 1, 77-95, doi:10.1016/j.ejor.2004.03.012

[6] Framinan, J. M. (2007). An adaptive branch and bound approach for transforming job shops into flow shops, *Computers & Industrial Engineering*, Vol. 52, No. 1, 1-10, doi:10.1016/j.cie.2006.10.001

[7] Saadani, N. E. H.; Guinet, A.; Moalla, M. (2005). A travelling salesman approach to solve the F/no-idle/C_{max} problem, *European Journal of Operational Research*, Vol. 161, No. 1, 11-20, doi:10.1016/j.ejor.2003.08.030

[8] Rathinam, B.; Govindan, K.; Neelakandan, B.; Raghavan, S. S. (2015). Rule based heuristic approach for minimizing total flow time in permutation flow shop scheduling, *Technical Gazette*, Vol. 22, No. 1, 25-32, doi:10.17559/TV-20130704132725

[9] Wenqi, H.; Aihua, Y. (2004). An improved shifting bottleneck procedure for the job shop scheduling problem, *Computers & Operations Research*, Vol. 31, No. 12, 2093-2110, doi:10.1016/S0305-0548(03)00243-0

[10] Tavakkoli-Moghaddam, R.; Daneshmand-Mehr, M. (2005). A computer simulation model for job shop scheduling problems minimizing makespan, *Computers & Industrial Engineering*, Vol. 48, No. 4, 811-823, doi:10.1016/j.cie.2004.12.010

[11] Vinod, V.; Sridharan, R. (2011). Simulation modeling and analysis of due-date assignment methods and scheduling decision rules in a dynamic job shop production system, *International Journal of Production Economics*, Vol. 129, No. 1, 127-146, doi:10.1016/j.ijpe.2010.08.017

[12] Peng, B.; Lü, Z.; Cheng, T. C. E. (2015). A tabu search/path relinking algorithm to solve the job shop scheduling problem, *Computers & Operations Research*, Vol. 53, 154-164, doi:10.1016/j.cor.2014.08.006

[13] Ponsich, A.; Coello Coello, C. A. (2013). A hybrid differential evolution-tabu search algorithm for the solution of job-shop scheduling problems, *Applied Soft Computing*, Vol. 13, No.1, 462-474, doi:10.1016/j.asoc.2012.07.034

[14] Nidhiry, N. M.; Saravanan, R. (2014). Scheduling optimization of a flexible manufacturing system using a modified NSGA-II algorithm, *Advances in Production Engineering & Management*, Vol. 9, No. 3, 139-151, doi:10.14743/apem2014.3.183

[15] Huang, X. W.; Zhao, X. Y.; Ma, X. L. (2014). An improved genetic algorithm for job-shop scheduling problem with process sequence flexibility, *International Journal of Simulation Modelling*, Vol. 13, No. 4, 510-522, doi:10.2507/IJSIMM13(4)CO20

[16] Elmi, A.; Solimanpur, M.; Topaloglu, S.; Elmi, A. (2011). A simulated annealing algorithm for the job shop cell scheduling problem with intercellular moves and reentrant parts, *Computers & Industrial Engineering*, Vol. 61, No. 1, 171-178, doi:10.1016/j.cie.2011.03.007

[17] Fister Jr, I.; Yang, X.-S.; Fister, I.; Brest, J.; Fister, D. (2013). A brief review of nature-inspired algorithms for optimization, *Elektrotehniski vestnik*, Vol. 80, No. 3, 116-122

[18] Marichelvam, M. K.; Prabaharan, T.; Yang, X. S. (2014). Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan, *Applied Soft Computing*, Vol. 19, 93-101, doi:10.1016/j.asoc.2014.02.005

[19] Li, J.-Q.; Pan, Q.-K; Tasgetiren, M. F. (2014). A discrete artificial bee colony algorithm for the multi-objective flexible job-shop scheduling problem with maintenance activities, *Applied Mathematical Modelling*, Vol. 38, No. 3, 1111-1132, doi:10.1016/j.apm.2013.07.038

[20] Zhang, R.; Song, S.; Wu, C. (2013). A hybrid artificial bee colony algorithm for the job shop scheduling problem, *International Journal of Production Economics*, Vol. 141, No. 1, 167-178, doi:10.1016/j.ijpe.2012.03.035

[21] Weckman, G. R.; Ganduri, C. V.; Koonce, D. A. (2008). A neural network job-shop scheduler, *Journal of Intelligent Manufacturing*, Vol. 19, No. 2, 191-201, doi:10.1007/s10845-008-0073-9

[22] Yang, S.; Wang, D. (2001). A new adaptive neural network and heuristics hybrid approach for job-shop scheduling, *Computers & Operations Research*, Vol. 28, No. 10, 955-971, doi:10.1016/S0305-0548(00)00018-6

[23] Sabuncuoglu, I.; Bayiz, M. (2000) Analysis of reactive scheduling problems in a job shop environment, *European Journal of Operational Research*, Vol. 126, No. 3, 567-586, doi:10.1016/S0377-2217(99)00311-2

[24] Jovanovic, J. R.; Milanovic, D. D.; Djukic, R. D. (2014). Manufacturing cycle time analysis and scheduling to optimize its duration, *Strojniski vestnik – Journal of Mechanical Engineering*, Vol. 60, No. 7-8, 512-524, doi:10.5545/sv-jme.2013.1523

[25] Gracanin, D.; Lalic, B.; Beker, I.; Lalic, D.; Buchmeister, B. (2013). Cost-time profile simulation for job shop scheduling decisions, *International Journal of Simulation Modelling*, Vol. 12, No. 4, 213-224, doi:10.2507/IJSIMM12(4)1.237

[26] Tecnomatix Plant Simulation, Siemens PLM Software, from *http://www.emplant.com/*, accessed on 10-02-2015

[27] Debevec, M.; Pipan, M.; Zupan, H.; Herakovic, N. (2016). An innovative training of production planners through virtual production performing, *Technical Gazette*, Vol. 23, In Press, doi:10.17559/TV-20150130124330

[28] eM-Plant (1998). *Reference Manual*, Tecnomatix Technologies GmbH & Co. KG, Stuttgart

[29] eM-Plant (1998). *Objects Manual*, Tecnomatix Technologies GmbH & Co. KG, Stuttgart

[30] Bangsow, S (2010). *Manufacturing Simulation with Plant Simulation and SimTalk*, Springer-Verlag, Berlin

[31] Yeh, J.-Y.; Lin, W.-S. (2007). Using simulation technique and genetic algorithm to improve the quality care of a hospital emergency department, *Expert Systems with Applications*, Vol. 32, No. 4, 1073-1083, doi:10.1016/j.eswa.2006.02.017