Research and Modification of the Universal Method of Scheduling and Operational Planning of Objects with a Network Representation of Discrete Type Production

Sergii Telenyk[†], **Alexander Pavlov**[††], **Elena Misura**[††], **Taras Lisetsky**[††], **Elena Khalus**[††], **and Oleg Melnikov**[††]

[†]Department of Theoretical Electrical Engineering and Computer Science, Cracow University of Technology, Warszawska 24, 31-155 Cracow, Poland
[††]Department of Computer-Aided Management and Data Processing Systems, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Peremohy ave. 37, 03056 Kyiv, Ukraine

Abstract
We study the efficiency of the universal method of scheduling and operational planning for objects with a network representation of discrete production processes (UMSOP) by means of statistical modeling. We claim the efficiency of the method in general and of using PSC-algorithms to solve NP-hard single-stage scheduling problems used in UMSOP. We present a modification of UMSOP, which consists in a proposition to use a new efficient approximation algorithm as a part of the operational planning model. The algorithm is created for the single-stage scheduling problem that allows to solve the operational plan adjusting problems more efficiently, comparing to the single-stage scheduling problem used in UMSOP. We show that the obtained solution is conditionally optimal and give a lower bound to the optimal value of this new problem's functional.

Key words:
Scheduling, Operational planning, Discrete production, PSC-algorithms, Approximation algorithm

1. Introduction

The problem of efficient planning in systems with a discrete nature of network-type production processes is of current attentive interest. This is connected to the facts that:

- discrete network-type production processes are the most common (for example, about four of five mechanical engineering enterprises are small-series productions);
- the problem of obtaining an operational plan which is optimal in terms of practical criteria for a production of such nature is in fact a multi-stage scheduling problem (MSSP) of a rather general type. As we know, obtaining efficient exact or approximation algorithms for this kind of problem is still a quite difficult problem today. Researches in this area are widely known, e.g., [1–5].

The purpose of this article is to substantiate experimentally the efficiency of the universal method of scheduling and operational planning of objects with a network representation of discrete production (UMSOP) and modify it using the theory of PSC-algorithms for single-stage intractable scheduling problems [6–9].

The paper contains two main parts. Section 2 presents the study of UMSOP's efficiency on the basis of a computer experiment. In Section 3, we give a modification of this method to increase its efficiency.

2. Study of UMSOP's Efficiency

UMSOP solves the following problem [5]. We need to build an operational plan to process a set of interrelated jobs at an object specified by a discrete network-type technological process. Optimization is based on five practical basic criteria given below and their arbitrary linear combinations. We also have to solve the operative planning problem which is the operational plan adjustment in case of its partial failure during execution. The jobs processing is considered within the network-type model. Its formal representation is a graph with several types of elements (Fig. 1 which is a shortened and modified version of Fig. 9.6 [5]).

In Fig. 1, an arrow entering a circle indicates a completed job, an arrow coming out of a circle is a new uncompleted job assigned for processing on the element the arrow points to; a square is a SMOJ-type element (single machine and one job). Other types of job processing elements: SMSJ (single machine and sequential jobs), PMCDD (identical parallel machines and jobs with a common due date), PMEP (identical parallel machines of equal productivities and jobs with arbitrary due dates), PMVP (identical parallel machines of various productivities and jobs with arbitrary due dates). A buffer denotes special precedence relations between jobs corresponding, for example, to assembling or disassembling procedures. C_i is the completion time of product i corresponding to the completion time of its terminal job [5].

The five basic optimality criteria are the following [5].

Manuscript received October 5, 2020
Manuscript revised October 20, 2020
https://doi.org/10.22937/IJCSNS.2020.20.10.5

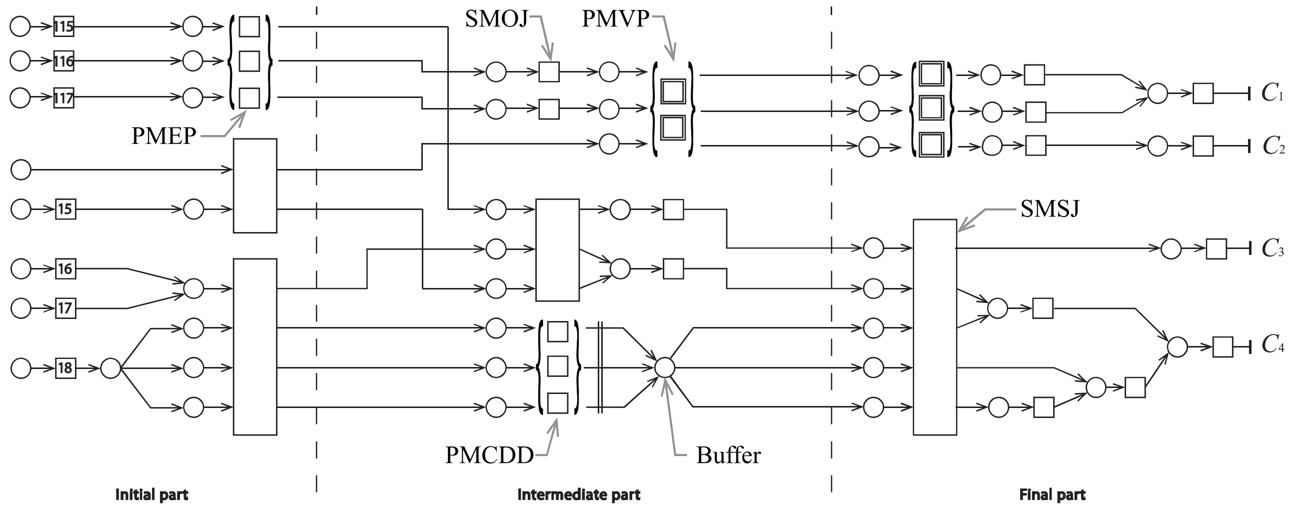

Fig. 1 A network model example with all types of elements.

Criterion 1 [5]. The enterprise's total profit maximization in the absence of the products' due dates:

$$F_1 = \max\left\{\sum_{i=1}^{n} \omega_i(T)\cdot(T - C_i)\right\} + P - E$$

$$\Leftrightarrow F_1 = \min \sum_{i=1}^{n} \omega_i(T)\cdot C_i .$$

Here, $\omega_i(T)$ is the weight coefficient of product i determined in an expert way; T is the planning period, $T \geq C_i, i = \overline{1,n}$; n is the number of products, P is the guaranteed minimum income from the sale of all n products; E is the total cost.

Criterion 2 [5]. The enterprise's total profit maximization subject to the condition: each product $i \in I$ has a due date d_i that must not be violated (just in time planning):

$$F_2 = \max\left\{\sum_{i=1}^{n} \omega_i U_i\right\}, \text{ where } U_i = \begin{cases} 1, C_i = d_i \\ 0, C_i \neq d_i \end{cases},$$

ω_i is the income from the sale of the product i if it completes just in time.

Criterion 3 [5]. The enterprise's total profit maximization subject to the conditions: each product $i \in I$ has a due date d_i; the total weighted tardiness of products in regard to their due dates must be minimized:

$$F_3 = \max\left\{P - \sum_{i=1}^{n} \omega_i \max(0, C_i - d_i)\right\} - E$$

$$\Leftrightarrow F_3 = \min\left\{\sum_{i=1}^{n} \omega_i \max(0, C_i - d_i)\right\},$$

where P is the guaranteed minimum income from the sale of all n products if they are not tardy; E is the total cost; ω_i is the cost of (penalty for) the tardiness of product i in relation to its due date per time unit. The value of $\omega_i \max(0, C_i - d_i)$ is the reduction of income P in the case if product i has the tardiness $C_i - d_i$.

Criterion 4 [5]. Each product $i \in I$ has a due date d_i and a given absolute value of profit ω_i for its processing. The profit does not depend on the completion time of the product if it is not tardy in regard to its due date. Otherwise, the enterprise's profit for this product is zero. The problem is to maximize the total profit of the enterprise:

$$F_4 = \max\left\{\sum_{i=1}^{n} \omega_i U_i\right\} - E \Leftrightarrow \max\left\{\sum_{i=1}^{n} \omega_i U_i\right\},$$

where $U_i = \begin{cases} 1, C_i \leq d_i \\ 0, C_i > d_i \end{cases}$, ω_i is the income from the sale of the product i if it is not tardy in regard to its due date, E is the risk of the profit loss due to the due dates violation.

Criterion 5 [5]. All products have due dates d_i. We need to minimize the total cost (penalty for the enterprise) both for earliness and tardiness in regard to the due dates:

$$F_5 = \max\left\{P - \sum_{i=1}^{n} \omega_i |C_i - d_i|\right\} - E \Leftrightarrow \min \sum_{i=1}^{n} \omega_i |C_i - d_i|,$$

where P is the guaranteed minimum income from the sale of all n products if they are not tardy or early; E is the total cost; ω_i is the cost of (penalty for) the deviation of the completion time of product i from the due date per time unit. The value of $\omega_i |C_i - d_i|$ is the reduction of income P in the case if product i has the tardiness $C_i - d_i$ or the earliness $d_i - C_i$.

The methodology of constructing the universal method UMSOP that solves the formulated problem is as follows [5]. A two-level procedure of the technological process aggregation was proposed on the basis of the new formal representation of discrete network-type technological processes. The second level of aggregation represents the initial production in the form of the following single-stage scheduling problem. A set of jobs is to be processed on a single machine taking into account the restrictions on the sequence imposed in the form of a directed acyclic graph. We need to find an optimal sequence that minimizes the total weighted completion time of jobs provided that the weights are specified only for the terminal jobs of products. This problem is called TWCTZ.

It is shown [5] that the first basic criterion leads to the TWCTZ problem at the second level of aggregation. An approximating TWCTZ problem is constructed for each of the basic criteria 2–5. The problem parameters are uniquely set by the parameters of the basic criterion.

The exact PSC-algorithm is used to solve the TWCTZ problem [10] (p. 354). The obtained optimal sequence of products (or product series) is used by the developed method of coordinated planning at the first level of the aggregated model. The result of coordinated planning are the completion times for the terminal jobs. Then the times are used as the due dates for the terminal jobs in the proposed formal model that represents discrete network-type production processes ($d_i = C_i$ in Fig. 1).

The resulting problem is a MSSP. Its goal is to obtain a feasible operational plan that minimizes the technological cycle time while implementing the given portfolio of order.

Thus, obtaining an optimal operational plan according to any of the five basic criteria or their arbitrary linear combination changes only the values of the due dates for the terminal jobs without changing the formulated MSSP in its substance.

The method for the MSSP solving is the appropriate procedure for an operational plan construction that starts from the terminal jobs of products and finishes with jobs at the beginning of the planning period. PSC-algorithms [11] for the following single-stage combinatorial optimization problems are used during the MSSP solving:

On SMSJ-type elements, we solve the problem of the total earliness minimization with arbitrary due dates of jobs on a single machine when jobs must not be tardy [7];

On PMCDD-type elements, we solve the total earliness minimization problem on identical parallel machines with a common due date of jobs that must not be tardy [8];

On PMEP-type elements, we solve the lexicographic or scalar criterion maximization problem on identical parallel machines of equal productivities with arbitrary due dates of jobs that must not be tardy [9];

On PMVP-type elements, we solve the lexicographic or scalar criterion maximization problem on identical parallel machines of various productivities with arbitrary due dates of jobs that must not be tardy [9].

The operative planning problem is to adjust the operational plan [5] in the final part of the network model (Fig. 1, where $d_i = C_i$). To solve this problem, in particular, PSC-algorithms are used for a number of single-stage combinatorial optimization problems [5] including the PSC-algorithm for the problem of independent jobs scheduling on identical parallel machines to minimize total tardiness when the machine start times are arbitrary but less than the common due date (TTPL) [12].

2.1. Methodology of the Efficiency Study of an Operational Plan Construction in UMSOP

As mentioned above, there are no algorithms to solve the formulated problem of scheduling and operational planning of objects with a network representation of discrete production according to the five specified practical basic criteria. Therefore, we propose to compare the efficiency of the universal method UMSOP with the “standard” method presented below. It is based on the following positions.

There is no other efficient way to solve the problem, except for the use of the aggregation/disaggregation methodology. Therefore, we propose to keep the procedures of aggregation and disaggregation, coordinated planning, the MSSP solving algorithm developed for the formal representation of network-type technological processes. But we exclude all PSC-algorithms for single-stage scheduling problems solving and replace them with efficient heuristic algorithms.

Thus, the statistical study of UMSOP’s efficiency actually reduces to proving the efficiency of using exact PSC-algorithms for solving NP-hard but partial single-stage scheduling problems used in UMSOP.

The “standard” method has such differences from the universal one. Instead of exact PSC-algorithms, we use efficient high-speed heuristic algorithms.

Firstly, for TWCTZ problem solving on an aggregated model of the second level of aggregation, we use the initial feasible sequence construction algorithm (see the algorithm for TWCTZ problem solving [5] (p. 481)) as the heuristic algorithm. It takes into account the priorities of products (the ratios of their weights and processing times) and the constraints imposed by the directed acyclic graph.

Secondly, we replace the algorithms for single-stage optimization problems solving on the network elements at the third level of UMSOP.

On SMSJ-type elements, we use the following heuristic algorithm to construct a schedule.

Algorithm A-SMSJ

1. Build a feasible (without tardy jobs) schedule in which jobs are in non-decreasing order of due dates [13] (Chap. 3).
2. Find the latest start time of jobs at which the obtained schedule remains feasible (see Algorithm A [7] (p. 18)).
3. Apply all feasible permutations in the obtained schedule to process longer jobs earlier. As shown in [7] (Theorem 2.3 on p. 21, Example 2.3 on p. 22), each permutation of such type reduces the total earliness of jobs.

On PMCDD-type elements, we use an algorithm based on sequential permutations that improve the functional value [8] starting from the schedule obtained by the greedy algorithm A02 [8].

On PMEP-type and PMVP-type elements, we use two heuristic algorithms for the scalar criterion (the earliest start time of machines maximization). Then we use the best of obtained solutions as the final result.

The first heuristic algorithm: we use Algorithm B2.1 from [9] (p. 50) on PMEP-type elements and Algorithm D2.1 from [9] (p. 78) on PMVP-type elements.

The second heuristic algorithm is the human-machine algorithm that combines efficient formal procedures with the professionalism of experts in the discussed subject:

1. Experts pre-assign jobs to the machines.
2. Algorithm A-SMSJ is applied on each machine.
3. Experts analyze the resulting assignment and reassign jobs to the machines if necessary, go to step 2.
4. Steps 2 and 3 are repeated until a feasible schedule is obtained that satisfies the experts.

Note. The second method is used for small and medium size problems.

2.2. Statistical Study of UMSOP

The research was done on a PC with an Intel's 3 GHz processor. We considered such dimensions (the number of operations in the network n): 100, 200, 500, 1000, 5,000, 10,000. 125 examples were generated for each dimension. Next, each of 750 examples was solved by both the universal and the "standard" method for each of the five basic optimization criteria (we did not test synthetic criteria, but solving common applied problems shows that mixed criteria usage has little effect on the average performance of UMSOP). Thus, we did 3750 runs for each method.

Table 1 below illustrates the statistical studies of UMSOP by showing the values of such indicators:

- $t^U_{\max} / t^U_{\min} / t^U_{\text{avg}}$ are the largest / the smallest / the average (for all 125 examples) time of the problem solving (obtaining an operational plan) by the *universal* method in seconds;
- $t^S_{\max} / t^S_{\min} / t^S_{\text{avg}}$ are the largest / the smallest / the average (for all 125 examples) time of the problem solving (obtaining an operational plan) by the *"standard"* method in seconds;
- $\Delta^S_{\max}$ / $\Delta^S_{\min}$ / Δ^S_{avg} are the largest / the smallest / the average (for all 125 examples) deviation of the functional value for the operational plan obtained by the "standard" method from that for the universal one, as a percentage of the functional value according to the universal method ("+" means a higher value for the "standard" method, "–" means a lower value).

The results show that, as a rule, the "standard" method runs 1.5 to 2 times faster than the universal one, but has 7 to 16 % worse efficiency in terms of the functional value.

3. The Modification of UMSOP

As we show in Section 2, solving the formulated problems of a production process planning is only possible through the use of the aggregation and disaggregation methodology. Thus, the modification of UMSOP consists in:

- modification of PSC-algorithms used in UMSOP, in order to increase their efficiency;
- creation of efficient algorithms for solving single-stage scheduling problems used in UMSOP, however in a more general formulation, which increases the efficiency of the method's application.

The modification of UMSOP presented in this article consists in generalization of the TTPL problem [12] (used at the fourth level of the model which is the operative planning level) and the proposed efficient method to solve the generalized problem.

3.1. TTPL Problem Description

Given a set of n tasks J and m identical parallel machines. We know a processing time l_j for each task $j \in J$. All tasks have a common due date d. We assume that a machine $i, i = \overline{1,m}$, can start to process any task of the set J after the time point s_i. The start times of the machines s_i satisfy $s_i < d$, $i = \overline{1,m}$, and may be different. The machines idle times are forbidden. We need to build such a schedule σ of the tasks $j \in J$ processing on m machines that minimizes the functional

$$f(\sigma) = \sum_{j \in J} \max\left(0, C_j(\sigma) - d\right)$$

where $C_j(\sigma)$ is the completion time of task j in the schedule σ [12].

Let us use the following notation and definitions [12]:

- $P_i(\sigma)$ is the set of non-tardy tasks in the schedule of machine i;
- $S_i(\sigma)$ is the set of tardy tasks in the schedule of machine i for which $C_j - l_j < d$, $C_j > d$, $\forall j \in S_i(\sigma)$, where $C_j - l_j$ is the start time of a task j;
- $Q_i(\sigma)$ is the set of tardy tasks in the schedule of machine i for which $C_j - l_j \geq d$, $\forall j \in Q_i(\sigma)$;
- $P = \bigcup_{i=\overline{1,m}} P_i$; $S = \bigcup_{i=\overline{1,m}} S_i$; $Q = \bigcup_{i=\overline{1,m}} Q_i$;
- $R_i(\sigma) = d - \sum_{j \in P_i(\sigma)} l_j$ is the time reserve of machine i in a schedule σ;

Table 1: Comparison of the universal and the "standard" method for the planning problem solving.

n	$t^U_{\max}$	$t^U_{\min}$	t^U_{avg}	$t^S_{\max}$	$t^S_{\min}$	t^S_{avg}	$\Delta^S_{\max}$	$\Delta^S_{\min}$	Δ^S_{avg}
The basic criterion 1									
100	0.29	0.10	0.15	0.25	0.08	0.10	+12.5	+8.4	+10.6
200	0.64	0.22	0.33	0.55	0.17	0.22	+13.0	+8.8	+9.8
500	2.57	0.88	1.28	2.17	0.66	0.86	+12.1	+9.5	+10.5
1,000	7.19	2.47	3.48	5.79	1.74	2.26	+13.7	+11.4	+12.2
5,000	129.43	44.71	61.09	97.45	29.07	37.26	+15.1	+8.9	+12.8
10,000	482.36	167.44	222.16	333.52	98.71	125.05	+14.9	+11.8	+12.6
The basic criterion 2									
100	0.84	0.30	0.48	0.74	0.23	0.46	+12.1	+9.2	+10.5
200	1.84	0.66	1.02	1.60	0.51	0.99	+13.0	+8.6	+11.0
500	7.67	2.77	4.14	6.38	1.99	3.85	+12.9	+7.5	+10.6
1,000	21.05	7.65	11.10	16.38	5.09	9.70	+13.5	+6.8	+10.8
5,000	392.87	143.44	202.15	280.71	86.49	163.00	+16.0	+7.0	+12.4
10,000	1,401.23	514.06	703.52	982.52	300.37	559.47	+19.7	+7.6	+14.9
The basic criterion 3									
100	1.45	0.57	0.93	1.28	0.43	0.82	+14.7	+8.7	+10.2
200	3.19	1.26	2.00	2.82	0.94	1.77	+13.8	+8.4	+9.8
500	13.07	5.19	7.98	10.80	3.57	6.66	+13.5	+8.4	+9.7
1,000	37.22	14.84	22.18	28.25	9.27	17.07	+13.8	+8.8	+10.1
5,000	664.71	266.40	386.49	495.16	161.25	293.35	+14.8	+9.7	+11.0
10,000	2,408.46	969.93	1366.48	1710.35	552.66	993.65	+15.9	+10.7	+12.0
The basic criterion 4									
100	0.55	0.21	0.38	0.51	0.17	0.31	+15.1	+9.5	+11.3
200	1.17	0.45	0.79	1.08	0.36	0.65	+13.5	+8.5	+9.9
500	4.96	1.91	3.26	4.23	1.39	2.47	+13.8	+8.6	+10.0
1,000	13.51	5.23	8.67	11.30	3.68	6.48	+14.9	+9.1	+10.5
5,000	245.16	95.43	153.56	195.49	63.17	109.89	+16.1	+9.8	+11.1
10,000	913.55	357.34	558.35	681.47	218.50	375.65	+16.0	+9.7	+10.9
The basic criterion 5									
100	0.78	0.37	0.55	0.77	0.25	0.44	+14.6	+8.3	+10.7
200	1.68	0.80	1.16	1.66	0.54	0.93	+14.2	+8.0	+10.5
500	6.85	3.28	4.60	6.64	2.12	3.65	+14.7	+8.2	+10.9
1,000	18.96	9.13	12.42	17.53	5.56	9.45	+15.5	+8.5	+11.6
5,000	353.85	171.13	226.20	306.00	96.31	161.71	+16.7	+9.1	+12.6
10,000	1,295.59	629.61	808.13	1,028.88	321.32	533.18	+16.0	+8.6	+12.1

- $\Delta_i(\sigma) = \sum_{j\in P_i(\sigma)\bigcup S_i(\sigma)} l_j - d$ is the tardiness of task $j \in S_i(\sigma)$ in regard to the due date;
- Ψ_{PS} is a class of schedules that correspond to the following conditions:
  1. $P\bigcup S = \{1, 2, \dots, |P\bigcup S|\}$ ($|P\bigcup S|$ is the cardinality of the set $P\bigcup S$);
  2. If $|P\bigcup S| < n$, then $\sum_{j\in P_i\bigcup S_i} l_j \ge d$ and Q_i contains those and only those elements which differ from $|P\bigcup S| + i$ by an amount that is a multiple of m, $i=\overline{1,m}$.
- $\Psi_P \subset \Psi_{PS}$ is a class of schedules satisfying the following additional conditions:
  1. $P = \{1, 2, \dots, |P|\}$;
  2. $\min_{j\in S(\sigma)} l_j > \max_{i=\overline{1,m}} R_i(\sigma)$;
  3. If $l_{j_k} \le l_{j_l}$, then $C_{j_k} - l_{j_k} \le C_{j_l} - l_{j_l}$ $\forall j_k, j_l \in S(\sigma)$.
- A schedule with the same number of tardy tasks on all machines is called an *even schedule*;
- $L_{\max}$ is the maximum number of tardy tasks on all machines;
- Numbers $i=\overline{1,k}$ correspond to machines with the number of tardy tasks $L_{\max}$;
- $\Delta_\Sigma(\sigma) = \sum_{i=1}^{k} \Delta_i$, $R_\Sigma(\sigma) = \sum_{i=k+1}^{m} R_i$;
- $\Omega_\Sigma(\sigma) = \min(R_\Sigma(\sigma), \Delta_\Sigma(\sigma))$.

It was shown in [14] that sufficient signs of optimality of a feasible solution derived for the case when the start times of machines are the same [6] are also valid for the case when the start times of machines are fixed and arbitrary.

Theorem 1 [12]. Sufficient sign of optimality of a feasible solution #1. An even schedule $\sigma \in \Psi_P$ is optimal.

Theorem 2 [12]. Sufficient sign of optimality of a feasible solution #2. If $\Omega_\Sigma(\sigma) = \min(R_\Sigma(\sigma), \Delta_\Sigma(\sigma)) = 0$ in a schedule $\sigma \in \Psi_P$, then the schedule σ is optimal.

The PSC-algorithm for the TTPL problem solving is given in [12] (pp. 305–312). It includes the polynomial component and the approximation algorithm and is built solely on directed permutations. After the problem solving, we obtain either a strictly optimal solution by the polynomial component of the algorithm (if any sufficient sign of optimality was satisfied during the computation) or an approximate one with an upper bound for the deviation from the optimum for every instance of the problem [12] (p. 312; Theorem 6.11 at p. 300).

3.2. Statistical Studies of the PSC-algorithm for TTPL Problem Solving

We generated random problem instances with dimensions ranging from 100 to 3,000 tasks and from 5 to 30 machines. We chose the processing times of tasks from a uniformly distributed interval of integers from 1 to 200, the chosen due date d was equal to 70 % of the total processing time of all tasks divided by the number of machines and rounded to the lower integer. The start times of the machines were chosen from a uniformly distributed interval of integers from 1 to d. We ran 2,000 runs for each (n, m) pair on a PC with an Intel's dual-core 3.4 GHz processor. We used the PSC-algorithm from [12] (p. 308) for the solution.

Average solving time (for all 2,000 runs) did not exceed 1 ms for the maximum size of the TTPL problem.

Table 2 shows the average (for all 2,000 runs) frequency of a sufficient sign of optimality fulfilment (an optimal solution obtaining) in percent depending on the problem size.

Statistical studies have shown high efficiency of the PSC-algorithm for TTPL problem solving. Thus, we achieved an optimal solution in 77.7 % of runs on average. The frequency of optimal solutions obtaining increases significantly with the introduction of additional types of permutations, and in that case it also increases with the problem size (up to 89–96 %) [6].

Table 2: Average frequency of an optimal solution obtaining, %

n \ m	5	7	10	15	20	30
100	94.7	100.0	98.4	73.0	65.0	95.3
200	83.6	96.8	97.1	86.0	97.5	74.9
400	83.8	75.7	84.3	85.5	99.9	99.2
600	79.7	82.1	91.2	84.3	90.5	99.4
800	70.3	81.0	81.8	92.8	89.7	90.8
1,000	72.0	69.8	64.6	94.8	91.7	71.1
1,500	65.2	67.9	68.2	65.0	71.3	93.9
2,000	61.0	60.6	65.7	80.2	48.5	96.7
2,500	63.3	63.2	67.7	54.8	50.6	79.0
3,000	65.6	58.0	64.5	65.4	57.0	39.1

3.3. Generalization of TTPL Problem

The generalized problem statement is based on TTPL problem statement [12]. Given a set of n tasks J and m identical parallel machines. We know a processing time l_j for each task $j \in J$. All tasks have a common due date d. We assume that a machine $i, i = \overline{1,m}$, can start to process any task of the set J after the time point s_i. The start times of the machines s_i satisfy $s_i < d$, $i = \overline{1,m}$, and may be different. The machines idle times are forbidden. Tasks $j \in J'$, $J' \subseteq J$, must complete without tardiness: $C_j \le d, j \in J'$. We need to build such a schedule σ of the tasks $j \in J$ processing on m machines that minimizes the functional

$$f(\sigma) = \sum_{j \in J} \max(0, C_j(\sigma) - d)$$

where $C_j(\sigma)$ is the completion time of task j in the schedule σ.

This problem (let us call it TTPLG) is studied for the first time. It is NP-hard since the TTPL problem is also NP-hard. This is a natural generalization of the TTPL problem, a single-stage scheduling problem used at the fourth level of UMSOP. We present an efficient approximation algorithm for its solution and show that there exists a constructively built lower bound of the optimal functional value for this algorithm.

3.4. The Approximation Algorithm for TTPLG Problem Solving

1. First, distribute (assign to mathines) tasks $j \in J'$ using the PSC-algorithm for the TTPL problem [12] (p. 305–312). The readiness times of the machines for tasks processing equal to s_i, $i = \overline{1,m}$.

   As a result of the problem solving by the PSC-algorithm, the following cases are possible:

   - It yields a feasible schedule (the total tardiness is zero). Go to step 2;
   - It yields an optimal solution (one of the sufficient signs of optimality is fulfilled during the solving), in which the total tardiness is greater than zero. In this case, there is no feasible solution for the original TTPLG problem (see Statement 1). The end of the algorithm;
   - a feasible solution has not been built, although it may exist. The end of the algorithm (the approximation algorithm has not solved the problem instance).

2. After assigning the tasks $j \in J'$ to machines, determine new readiness times of the machines for tasks processing. Exclude tasks $j \in J'$ from the set J. The formulated problem is a TTPL problem. Solve it with the PSC-algorithm from [12].

3.5. Theoretical Properties of the Approximation Algorithm

Statement 1. Suppose that the PSC-algorithm for the TTPL problem formulated at Step 1 of the approximation algorithm has found an optimal solution to the problem (its polynomial component has built a solution that satisfies one of the sufficient signs of optimality), and the total tardiness in this solution is strictly greater than zero. Then, the original TTPLG problem has no feasible solution.

Proof. Let a feasible solution to the original problem exist. Then, there is a feasible solution with zero total tardiness to the problem formulated at Step 1. But this is impossible, since the value of the total tardiness for an arbitrary solution of the problem formulated at Step 1 cannot be less than the total tardiness of its optimal solution. ■

Statement 2. Suppose that the PSC-algorithm for solving the TTPL problem formulated at Step 2 of the approximation algorithm has solved the obtained problem instance optimally (its polynomial component has built a solution that satisfies one of the sufficient signs of optimality). Then, the resulting schedule obtained after Steps 1 and 2 execution is conditionally optimal: we have got the minimal total tardiness of tasks from the set $J \setminus J'$ for the readiness times of machines found at Step 2 for processing the tasks from this set.

Validity of Statement 2 is obvious.

Statement 3. Let us remove the restriction of completion the tasks $j \in J'$ without tardiness in the TTPLG problem. Solve the obtained TTPL problem with the PSC-algorithm [12]. Then, the following is true for the optimal solution of the problem (if one of the sufficient sign optimality was satisfied during the problem solving):

- if all tasks of the set J' do not violate the due date, then an optimal solution to the TTPLG problem is obtained;
- if at least one task $j \in J'$ violates the due date, then the functional value in the optimal solution of the TTPL problem formulated in Statement 3 is the lower bound to the functional value in the optimal solution of the TTPLG problem.

Proof follows from the fact that the optimal functional value of the TTPL problem after removing the restrictions on the tasks of the set J' cannot be greater than the optimal functional value of the original problem. ■

4. Conclusions

1. We have substantiated the current interest in operational plans building for objects with a network representation of discrete production.
2. We have substantiated and presented the methodology of studying the efficiency of UMSOP [5, 6].
3. We have presented the results of statistical studies of UMSOP.
4. We have substantiated the necessity of using the theory of PSC-algorithms to solve NP-hard single-stage scheduling problems used in the general scheme for an operational plan construction and its adjustment during its execution.
5. We have proposed a modification of the UMSOP based on the use of a new single-stage scheduling problem and an efficient method for its solving at the fourth level of UMSOP's implementation (the level of the operational plan adjustment). We have presented an approximation algorithm for the new problem solving and a constructive theoretical study of the solution obtained by the approximation algorithm.

References

[1] A. A. Pavlov and S. F. Telenyk, Informatsionnye tekhnologii i algoritmizatsiya v upravlenii (Informational technologies and algorithmization in control), Tekhnika, Kyiv, 2002 (in Russian).

[2] D. Ivanov, A. Dolgui, and B. Sokolov, "A dynamic approach to multi-stage job shop scheduling in an Industry 4.0-based flexible assembly system," presented at Advances in Production Management Systems. The Path to Intelligent, Collaborative and Sustainable Manufacturing: APMS 2017. Springer, Cham. DOI: 10.1007/978-3-319-66923-6_56

[3] J. Xu, S. Zhang and Y. Hu, "Research on construction and application for the model of multistage job shop scheduling problem,". Mathematical Problems in Engineering, Vol. 2020, Article ID 6357394, pp. 1-12, 2020. DOI: 10.1155/2020/6357394

[4] J. Blazewicz, K.H. Ecker, E. Pesch, G. Schmidt, M. Sterna, and J. Weglarz, Handbook on Scheduling: From Theory to Practice, 2nd ed., Springer, Cham, 2019. DOI: 10.1007/978-3-319-99849-7

[5] M. Z. Zgurovsky and A. A. Pavlov, "Algorithms and software of the four-level model of planning and decision making," in Combinatorial Optimization Problems in Planning and Decision Making: Theory and Applications, 1st edn., Springer, Cham, 2019. DOI: 10.1007/978-3-319-98977-8_9

[6] M. Z. Zgurovsky and A. A. Pavlov, Prinyatie resheniy v setevykh sistemakh s ogranichennymi resursami (Decision making in network systems with limited resources), Naukova dumka, Kyiv, 2010 (in Russian)

[7] M. Z. Zgurovsky and A. A. Pavlov, "Optimal scheduling for two criteria for a single machine with arbitrary due dates of tasks," in Combinatorial Optimization Problems in Planning and Decision Making: Theory and Applications, 1st edn., Springer, Cham, 2019. DOI: 10.1007/978-3-319-98977-8_2

[8] A. A. Pavlov and M. O. Sperkach, "Vypolnenie zadaniy s obshchim direktivnym srokom parallel'nymi priborami po kriteriyam optimal'nosti: minimizaciya summarnogo operezheniya otnositel'no direktivnogo stroka i maksimizaciya momenta zapuska zadaniy na vypolnenie"

(Tasks execution with a common due date on parallel machines with optimality criteria: total earliness minimization regarding the due date and the tasks' start time execution maximization), Visnyk NTUU KPI Inform. Oper. Comput. Sci., Vol. 62, pp. 89–92, 2015 (in Russian).

[9] M. Z. Zgurovsky and A. A. Pavlov, "Optimal scheduling for vector or scalar criterion on parallel machines with arbitrary due dates of tasks," in Combinatorial Optimization Problems in Planning and Decision Making: Theory and Applications, 1st edn., Springer, Cham, 2019. DOI: 10.1007/978-3-319-98977-8_3

[10] M. Z. Zgurovsky and A. A. Pavlov, "The total weighted completion time of tasks minimization with precedence relations on a single machine," in Combinatorial Optimization Problems in Planning and Decision Making: Theory and Applications, 1st edn., Springer, Cham, 2019. DOI: 10.1007/978-3-319-98977-8_7

[11] M. Z. Zgurovsky and A. A. Pavlov, "Introduction," in Combinatorial Optimization Problems in Planning and Decision Making: Theory and Applications, 1st edn., Springer, Cham, 2019. DOI: 10.1007/978-3-319-98977-8_1

[12] M. Z. Zgurovsky and A. A. Pavlov, "The total tardiness of tasks minimization on identical parallel nachines with arbitrary fixed times of their start and a common due date," in Combinatorial Optimization Problems in Planning and Decision Making: Theory and Applications, 1st edn., Springer, Cham, 2019. DOI: 10.1007/978-3-319-98977-8_6

[13] V. S. Tanaev and V. V. Shkurba, Vvedenie v Teoriju Raspisaniy (Introduction to Scheduling Theory), Nauka, Moscow, 1975 (in Russian).

[14] A. A. Pavlov, E. B. Misura, and T. N. Lisetsky, "Sostavlenie raspisaniya vypolneniya nezavisimyh zadanii identichnymi parallelnymi priborami, momenty zapuska kotoryh menshe obschego direktivnogo sroka" (Scheduling independent tasks on the identical parallel machines which starting times are smaller than a common due date). Visnyk NTUU KPI Inform. Oper. Comput. Sci., Vol. 58, pp. 24–28, 2013 (in Russian).

Sergii Telenyk graduated in 1975 from Igor Sikorsky Kyiv Polytechnic Institute, Ukraine, which is connected with his further scientific and educational activity. He has got PhD degree in 1982 and Doct. of Science degree in 2000. Professor and head of the Dept. of Automation and Control in Technical Systems since 2001. Dean of the Faculty of Informatics and Computer Engineering since 2018. Winner of the State Prize of Ukraine in Science and Technology (2018). He is currently a professor of Dept. of Theoretical Electrical Engineering and Computer Science, Cracow University of Technology. His research interests include models and methods of IT infrastructure management, design and programming automation, mathematical logic, computational linguistics, artificial intelligence, control theory, information systems design.

Alexander Pavlov graduated in 1967 from Igor Sikorsky Kyiv Polytechnic Institute, Ukraine, and connected with it his further scientific and educational activity. He has got PhD degree in 1970 and Doct. of Science degree in 1977. Associate professor of Technical Cybernetics Dept. since 1974. Professor (since 1979) and head (since 1981) of the Dept. of Computer-Aided Management Systems (renamed to Dept. of Computer-Aided Management and Data Processing Systems in 1994). Dean of the Faculty of Informatics and Computer Engineering in 1995–2018. Winner of the State Prize of Ukraine in Science and Technology (2009). Academician in Ukrainian Academy of Science of High School. His research interests include stability theory of nonlinear dynamic deterministic and stochastic control systems, operations research, intractable combinatorial and scheduling problems, models and methods of planning and control. He is a member of the Expert Council of the Accreditation Commission of Ukraine in the Ministry of Education and Science of Ukraine.

Elena Misura graduated from Kyiv Institute of Food Technologies, Ukraine, in 1964 and from Igor Sikorsky Kyiv Polytechnic Institute (KPI) in 1973. Her further scientific activity is connected with KPI. She has got PhD degree in 1981. She was an associate professor and head substitute of the Dept. of Computer-Aided Management and Data Processing Systems. Senior researcher since 1991. Her research interests include operations research, scheduling, and intractable combinatorial problems solution.

Taras Lisetsky graduated from Igor Sikorsky Kyiv Polytechnic Institute, Ukraine, in 2010 and its post-graduate school in 2013. He is now an entrepreneur for software engineering in Kyiv and applies for PhD degree in the Dept. of Computer-Aided Management and Data Processing Systems. His research interests include software design, scheduling, APS software.

Elena Khalus graduated from Igor Sikorsky Kyiv Polytechnic Institute, Ukraine, in 2004 and its post-graduate school in 2007. She is now a senior lecturer in Igor Sikorsky KPI and applies for PhD degree in the Dept. of Computer-Aided Management and Data Processing Systems. Her research interests include algorithms theory, operations research, scheduling, software design.

Oleg Melnikov graduated in 1995 from Igor Sikorsky Kyiv Polytechnic Institute, Ukraine, Dept. of Computer-Aided Management and Data Processing Systems, and connected with it his further scientific activity. He has got PhD degree in 2013 and is now a researcher in the department mentioned. His research interests include intractable scheduling problems, models and methods of planning and control, APS software.