Applied Computer Science, vol. 11, no. 4, pp. 58–69

Submitted: 2015-09-13
Revised: 2015-10-30
Accepted: 2015-12-01

production scheduling, deterministic scheduling, LiSA software

Łukasz SOBASZEK[*], *Arkadiusz GOLA*[**]

COMPUTER-AIDED PRODUCTION TASK SCHEDULING

Abstract

The following paper is devoted to computer-aided production scheduling. The initial presentation of principles of deterministic scheduling was followed by the description of typical production environments and completed by the classification of production tasks scheduling methods. Furthermore, LiSA software was introduced and applied to build a schedule based on actual production data. In conclusion, the effectiveness of production task scheduling was evaluated with selected logarithms offered by LiSA software.

1. INTRODUCTION

Modern, highly competitive market exerts constant pressure on production companies to meet the growing demands of customers [12]. To keep pace with the market, it is vital to establish the job processing capability of a particular company, which can be aided by developing the task schedule. Devising such a schedule can, nevertheless, often prove somewhat problematic [6]. A varying degree of processes complexity and availability of resources are two of numerous potential constraints contributing to the complexity of job scheduling, thus making it quite a time-consuming task. With the purpose of facilitating the production process, planning computer software for production scheduling is employed. Unfortunately, commercially available production task scheduling software is burdened with numerous limitations [16]. It is owing to that fact that great effort is put into developing effective scheduling algorithms.

[*] Lublin University of Technology, Institute of Technological Systems of Information, 20-618 Lublin, Nadbystrzycka 36, e-mail: l.sobaszek@pollub.pl
[**] Lublin University of Technology, Department of Enterprise Organization, 20-618 Lublin, Nadbystrzycka 38, e-mail: a.gola@pollub.pl

New generation of software, represented by LiSA package, enables devising job scheduling solutions for different variants.

2. DETERMINISTIC TASK SCHEDULING

Conducting the computer-aided process of task scheduling requires proper representation of the analysed production system in the form required by LiSA software. Hence, developed models of production environments contain certain assumptions and simplifications [3]. Computer-aided scheduling demands that the analysed and sequenced processes be deterministic, *i.e.* allowing for no chance variation. This implies that all parameters are known and fixed [10].

2.1. Types of production systems

Sequencing and scheduling of production consists in distributing tasks (involving certain number of operations) between *processors*, *i.e.* machines realising a particular process [7]. There are three basic models of production systems approached in the theory of task scheduling [10, 14]:

– *flow shop,*
– *job shop,*
– *open shop.*

In a flow-shop system, which is represented by an assembly line, the order of tasks realised on all machines is identical. Specialist literature frequently describes the problem of *permutation flow shop*, where the permutation of the set of tasks defines the decision variable [9].

Job shop is one of the most widespread systems for task scheduling. In this system, the sequencing of tasks is determined by technological constraints: the machine route is fixed and the job order can be selected. Frequently, such environments are referred to as *general job-shop* [17].

Open-shop systems are rarely found in the theory of work scheduling. What is characteristic of these types of environments is that the order of jobs is arbitrary [5]. The lack of pre-defined sequence of job orders determines that finding the optimal solution is highly complicated, particularly in terms of processing time. The space of calculations and searching for solution expands on account of a substantial number of diverse schedule variants [14].

2.2. Job-shop

The job-shop model is the closest representation of a typical production environment found in the mechanical engineering industry, hence the subsequent part of the paper will focus on general job-shop scheduling problem.

Typical solutions to job-shop scheduling problems found in literature are based on a set of following simplifications [3]:

1. No two tasks of the same job can be scheduled in parallel.
2. Each machine is capable of carrying out one task at a time.
3. Each job has a limited number of operations of one per machine.
4. Each task is processed completely.
5. Task processing time can be entered manually.
6. Waiting time between two successive operations is allowed.
7. There are no two identical machines.
8. Machines can be idle. Machines perform one task at a time.
9. No machine failure. Machines are available throughout the whole manufacturing process.
10. All technological constraints are known and fixed. No variation is allowed.
11. No alternative process plans are allowed.

The following assumptions might or might not be reflected in reality, nevertheless, their introduction is essential to developing a scheduling model for the production process. It is only through simplification of an existing production system that computer-aided production task scheduling can be conducted and its optimisation evaluated.

3. TASK SCHEDULING METHODS

Development of a production schedule is frequently a complex computational challenge. The literature discusses cases where task scheduling becomes an NP-hard problem [15, 18, 4]. NP-hard problem of scheduling involves long execution time of algorithm. The complexity of solution increases with the growing number of performed tasks, as well as with the growing number of machines carrying out the production process. The only effective optimal algorithms developed so far could solve this problem for one- and two-machine problems (*e.g.* Johnson algorithm [11]). That is why, there is an ongoing search for diverse scheduling methods, based on effective scheduling algorithms [13].

In general, scheduling solutions can be classified under two categories [5]:

– exact – guaranteeing determination of an optimal solution,
– approximate – the determined solution is non-optimal, however, the solution is delivered in a substantially smaller running time and requires engagement of fewer resources.

In terms of task scheduling methods the following could be mentioned [10]:

– Full and Random Search,
– Discrete Programming,

– Branch & Bound Algorithm,
– Expert Systems,
– heuristic scheduling methods (Dispatch Systems, Priority Rules, Local Search),
– evolutionary algorithms (Genetic Algorithms, Evolutionary Programming, Classification Systems).

4. LiSA SOFTWARE

At present, production scheduling processes are widely supported by a number of computer programmes. One drawback of production scheduling modules is their limited capabilities and flexibility, which proves to be a significant disadvantage, owing to the fact that frequently different job sequencing variants must be analysed. This often involves the change of the character of processed tasks or the objective function; furthermore, total completion time at different priority rules is analysed.

It is, *inter alia*, for the aforementioned reasons that LiSA software is a practical solution for solving deterministic problems of task scheduling. The name is an acronym, which stands for *Library of Scheduling Algorithms*. The programme is equipped with an uncomplicated and intuitive interface, shown in Fig. 1.

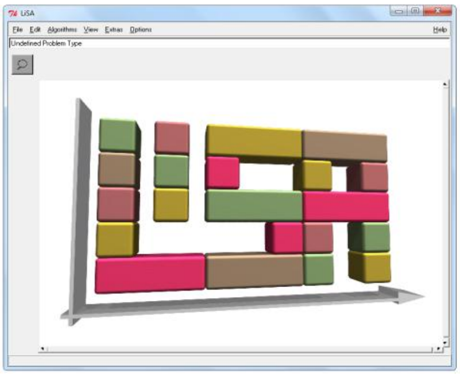

Fig. 1. LiSA – main interface [source: own study]

The user easily inputs data regarding the production process, which are subsequently inserted in suitable matrices; what is more, the application offers random generation of data. All data is gathered in XML files, hence the access to input data as well as results is facilitated. The work with LiSA programme can be stopped at any time, saved and continued when required. To conduct a computer-aided process of job scheduling it is required that the number of jobs and machines is defined, along with other essential data, such as [1, 2]:

– processing times of particular jobs – in matrix of processing times *PIJ*:

$$PIJ = [p_{ij}], \tag{1}$$

where: p_{ij} – processing time of job i on machine j.

– machine orders – in matrix of machine orders *MO*:

$$MO = [o_{ij}], \tag{2}$$

where: o_{ij} – rank of jobs i on machine j.

– set of operations – in matrix of set of operation *SIJ*:

$$SIJ = [m_{ij}], \tag{3}$$

$$m_{ij} = \{0,1\},$$

where: m_{ij} – information regarding processing of operations i on machine j (0 – operation is not processed, 1 – operation is processed).

LiSA uses the classification for deterministic scheduling problems called the Graham, or $\alpha \mid \beta \mid \gamma$ three-field, notation (Fig. 2), where α describes machine environment, β – constraints, γ – the objective function. Such notation enables clear description of a given task-scheduling problem [8].

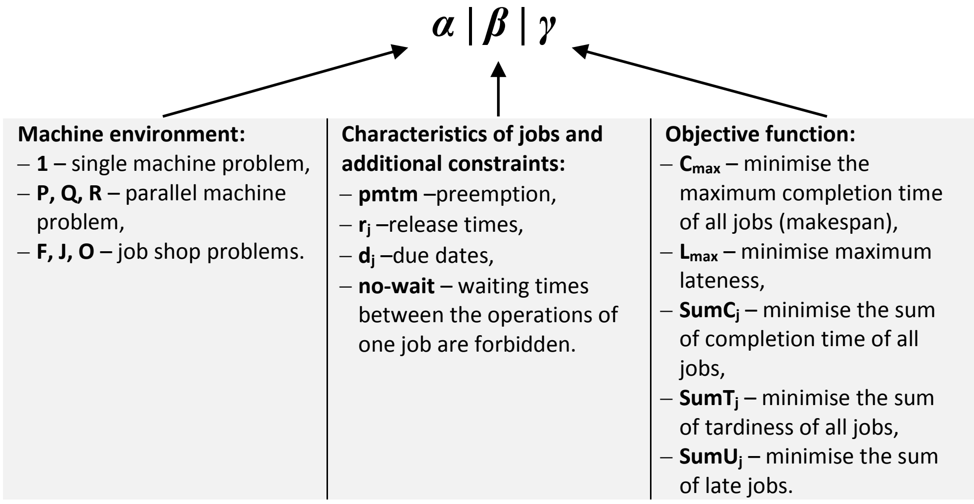

Fig. 2. The three-field Graham notation used by LiSA software [source: own study]

LiSA software solves deterministic task scheduling problems using its library of algorithms. The programme, furthermore, enables modification of existing solutions and implementation of new ones. Determining a job-shop schedule by means of LiSA package can be conducted with the use of the following methods [2]:

– Branch & Bound,
– Brucker's Job-Shop B&B,
– Dispatching Rules,
– Shifting Bottleneck
– Iterative Improvement,
– Simulated Annealing,
– Threshold Accepting,
– Tabu Search.

Each method allows modification of algorithm parameters in order to change their impact on effectiveness of scheduling. Any given problem can be modelled in the form of rank matrix (sequence), disjunctive graph or the Gantt chart (Fig. 3).

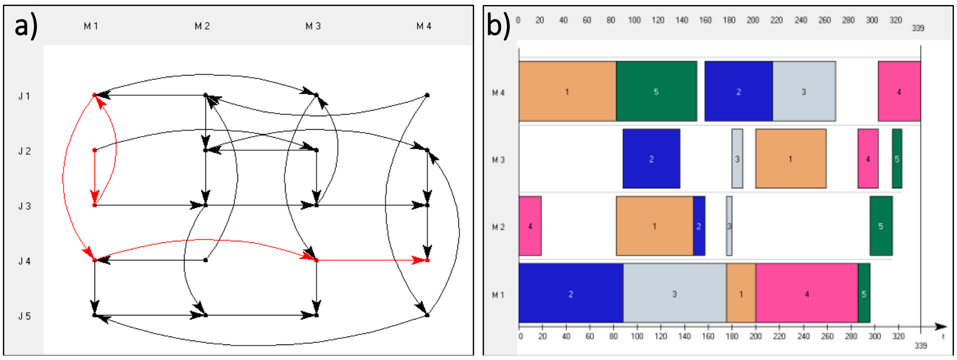

Fig. 3. Models for shop problems in LiSA: a) disjunctive graph, b) Gantt chart [source: own study]

5. EFFECTIVENESS OF TASK SCHEDULING ALGORITHMS

The evaluation of task scheduling algorithms used by LiSA software was based on actual manufacturing data. Four technological processes carried out in a machinery park consisting of 13 machine tools (the machines were ordered according to technology). In order to carry out the analysis, the data had to be suitably prepared, *i.e.* proper indices were ascribed to particular processes (jobs), machines and operations. The indices corresponded to the numbers used in LiSA software. All data is shown in Tables 1–4.

Tab. 1. Route sheet for process 1 [source: own study]

Technological process 1 **Name in LiSA:** job J1								
Operation No.	Machine	**Operation No. (LiSA)**	**Machine No. LiSA**	Operation	t_s [h]	t_c [h]	$t_s + t_c$ [h]	t for 100 units
10	WCC80	**1**	**1**	Centring and facing	0.3	0.03	0.33	3.3
20	TUM25	**2**	**4**	Turning	0.4	0.55	0.95	55.4
30	FYC26	**3**	**6**	Slotting	0.5	0.04	0.54	4.5
40	WKA25	**4**	**9**	Drilling	0.4	0.04	0.44	4.4
50	Met. work.	**5**	**10**	Metal working operations	0.15	0.12	0.27	12.15
60	TUD40	**6**	**3**	Finish turning	0.5	0.15	0.65	15.5
70	SWB25	**7**	**12**	Grinding	0.5	0.15	0.65	15.5

Table 2. Route sheet for process 2 [source: own study]

Technological process 2 **Name in LiSA:** job J2								
Operation No.	Machine	**Operation No. (LiSA)**	**Machine No. LiSA**	Operation	t_s [h]	t_c [h]	$t_s + t_c$ [h]	t for 100 units
10	WCC80	**1**	**1**	Centring and facing	0.3	0.03	0.33	3.3
20	TUD50	**2**	**2**	Turning	0.4	0.12	0.52	12.4
30	FWD25	**3**	**7**	Slotting	0.3	0.02	0.32	2.3
40	FYC26	**4**	**6**	Slotting	0.5	0.01	0.51	1.5
50	SWB25	**5**	**12**	Grinding	0.5	0.2	0.7	20.5
60	TUD40	**6**	**3**	Thread turning	1.5	0.18	1.68	19.5

Table 3. Route sheet for process 3 [source: own study]

Technological process 3 **Name in LiSA:** job J3								
Operation No.	Machine	**Operation No. (LiSA)**	**Machine No. LiSA**	Operation	t_s [h]	t_c [h]	$t_s + t_c$ [h]	t for 100 units
10	WCC80	**1**	**1**	Centring and facing	0.5	0.08	0.58	8.5
20	PHW12S	**2**	**5**	Turning	0.15	0.2	0.35	20.15
30	TUD50	**3**	**2**	Slotting	0.4	0.1	0.5	10.4
40	TUD50	**4**	**2**	Slotting	0.4	0.11	0.51	11.4
50	TUD50	**5**	**2**	Grinding	0.4	0.25	0.65	25.4
60	TUD40	**6**	**3**	Thread turning	1.5	0.45	1.95	46.5
70	FYC26	**7**	**6**	Slotting	0.5	0.08	0.58	8.5
80	SZX160LC	**8**	**11**	Grinding centring holes	0.2	0.02	0.22	2.2
90	SWB25	**9**	**12**	Grinding	0.5	0.29	0.79	29.5
100	Matrix	**10**	**13**	Grinding thread	1.5	1.9	3.4	191.5

Table 4. Route sheet for process 4 [source: own study]

Technological process 4 **Name in LiSA:** job J4								
Operation No.	Machine	**Operation No. (LiSA)**	**Machine No. LiSA**	Operation	t_s [h]	t_c [h]	$t_s + t_c$ [h]	t for 100 units
10	WCC80	**1**	**1**	Milling	0.4	0.03	0.43	3.4
20	TUD40	**2**	**3**	Turning	0.4	0.06	0.46	6.4
30	FND32	**3**	**8**	Milling	0.25	0.04	0.29	4.25
40	Met. work.	**4**	**10**	Metal working operations	0.15	0.01	0.16	1.15
50	SWB25	**5**	**12**	Grinding	0.5	0.18	0.68	18.5

5.1. Mathematical model

The collected data describing the realised processes and indexation allowed developing the mathematical description, which was subsequently implemented into LiSA software. The columns of the matrix represent processors (the number of technological machine) and rows denote the number of the job (realised technological process).

The mathematical description comprises the following elements:

– set of machines M:

$$M = \{M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{12}, M_{13}\}, \qquad (4)$$

– set of jobs J:

$$J = \{J_1, J_2, J_3, J_4\}, \qquad (5)$$

– matrix *SIJ*, containing information on the use of machines during particular production processes:

$$SIJ = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \qquad (6)$$

– matrix *PIJ*, containing processing times:

$$PIJ = \begin{bmatrix} 3 & 0 & 16 & 55 & 0 & 5 & 0 & 0 & 4 & 12 & 0 & 16 & 0 \\ 3 & 12 & 20 & 0 & 0 & 2 & 2 & 0 & 0 & 0 & 0 & 21 & 0 \\ 9 & 47 & 47 & 0 & 20 & 9 & 0 & 0 & 0 & 0 & 2 & 30 & 192 \\ 3 & 0 & 6 & 0 & 0 & 0 & 0 & 4 & 0 & 1 & 0 & 19 & 0 \end{bmatrix} \qquad (7)$$

– matrix *MO*, describing the order of operations on particular machines:

$$MO = \begin{bmatrix} 1 & 0 & 6 & 2 & 0 & 3 & 0 & 0 & 4 & 5 & 0 & 7 & 0 \\ 1 & 2 & 6 & 0 & 0 & 4 & 3 & 0 & 0 & 0 & 0 & 5 & 0 \\ 1 & 3 & 4 & 0 & 2 & 5 & 0 & 0 & 0 & 0 & 6 & 7 & 8 \\ 1 & 0 & 2 & 0 & 0 & 0 & 0 & 3 & 0 & 4 & 0 & 5 & 0 \end{bmatrix} \qquad (8)$$

The document containing mathematical model in the presented format can serve as input file for LiSA. The process of data input comprises the following steps:

1. In *Problem type* the user defines the analysed problem by means of three-field notation.
2. Then, in *Parameters* window the user describes the analysed process, by filling in the data for particular matrices (Fig. 4).
3. Next, a required scheduling algorithm is selected in the window *Algorithms*, where it can be moreover edited.
4. The obtained results are displayed after selecting *View*.

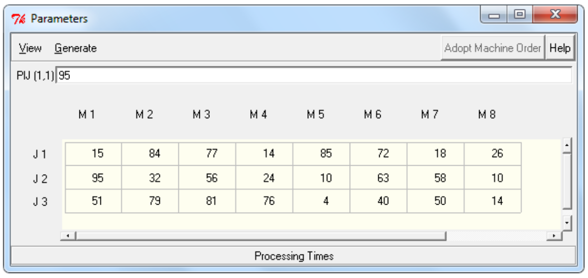

Fig. 4. Data input procedure – defining processing times [source: own study]

5.2. Effectiveness of selected production task scheduling algorithms

In order to evaluate the effectiveness of selected task scheduling algorithms, they were used to solve the following problem (the Graham notation):

$$J\,|\;|\,Cmax$$

The scheduled problem concerned the job shop system with no additional constraints, where the objective criterion was to minimise the schedule length. The latter is the criterion that is routinely introduced to analyse optimisation of production task scheduling algorithms.

One exact and three heuristic algorithms were applied in the analysed scheduling. The optimisation analysis involved assessing the impact of priority rules on scheduling results. The values parameters of implemented algorithms were unchanged and set to default. The results of scheduling are presented in Table 5.

Tab. 5. Results for task scheduling problem

Method	**Rule**	**C_{max} [h]**
Branch & Bound	LPT	356
	RANDOM	356
Dispatching Rules	LPT	428
	SPT	365
	FCFS	375
	RANDOM	359
Simulated Annealing	–	356
Tabu Search	–	356

In the analysed case, where the production process consists of four technological processes, methods based on both exact and approximate algorithms produced excellent effects. As a result, a 356 h schedule was determined.

As observed in Dispatching Rules the selection of priority rule has great impact on the schedule length. The difference between the longest and shortest schedule length amounts to 69 h. Fig. 5 shows Gantt chart for the optimal solution. The sequence of jobs is presented in the form of task scheduling matrix (Fig. 6). It ought to be noted that in this particular case the total schedule length depends heavily on the operations of technological process 3, which are characterised by long operation times. It is therefore justified to claim that the time of job 3 affects the realisation of other tasks.

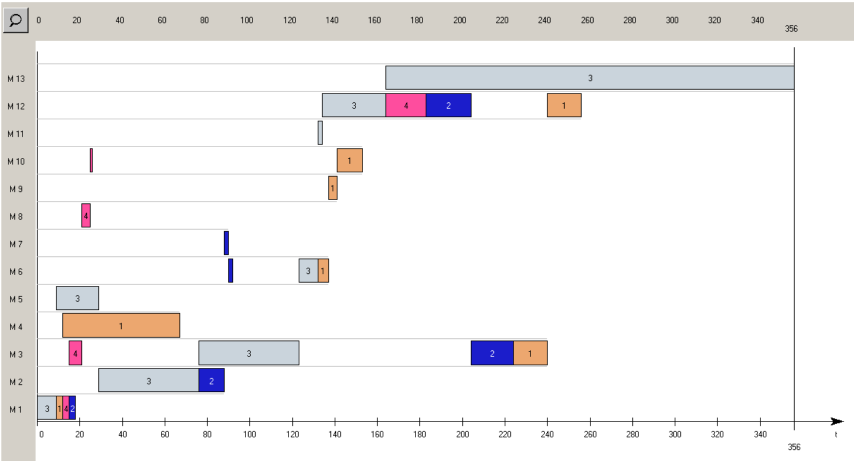

Fig. 5. Task schedule – Gantt chart [source: own study]

	M 1	M 2	M 3	M 4	M 5	M 6	M 7	M 8	M 9	M 10	M 11	M 12	M 13
J 1	2		14	3		9			10	11		15	
J 2	4	5	13			7	6					12	
J 3	1	3	5		2	8					9	10	11
J 4	3		4					5		6		11	

Fig. 6. Task scheduling matrix [source: own study]

6. SUMMARY

Production task scheduling is of great significance in the present market. Although designing a feasible schedule is frequently complex and problematic, a well-developed one allows tackling problems regarding order realisation. Hence, computer software is applied to aid production scheduling process. Computer-aided scheduling facilitates the work of any production plant and offers an array of solutions to different problem variants.

REFERENCES

[1] ANDRESEN M., BRÄSEL H., ENGELHARDT F., WERNER F.: *LiSA – A Library of Scheduling Algorithms – Handbook for Version 3.0*. Magdeburg University, 2001.

[2] BRÄSEL H., DORNHEIM L., KUTZ S., MÖRIG M., RÖSSLING I.: *LiSA – A Library of Scheduling Algorithms*. Magdeburg University, 2001.

[3] FRENCH S.: *Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shop*. J. Wiley & Sons, New York, 1982.

[4] GONZALEZ T., SAHNI S.: *Flowshop and Jobshop Schedules: Complexity and Approximation*. Operations Research, Vol. 26, No. 1, 1978, pp. 36–52.

[5] HERKA W., SEWASTIANOW P.: *Szeregowanie zadań na jednej maszynie (aspekt wąskiego gardła) w warunkach niepewności rozmyto-interwałowej*. III Seminarium „Metody matematyczne, ekonomiczne i informatyczne w finansach i ubezpieczeniach", Częstochowa, 20 listopad 2003 r.

[6] KŁOSOWSKI G.: *Zastosowanie symulacji komputerowej w sterowaniu przepływem produkcji mebli*. Zarzadzanie Przedsiębiorstwem, nr 2, 2011, s. 29–37.

[7] KNOSALA R.: *Zastosowania metod sztucznej inteligencji w inżynierii produkcji*. Wydawnictwa Naukowo-Techniczne, Warszawa, 2002.

[8] LEGRAND A.: *Introduction to Scheduling Theory*, 2004.

[9] NOWICKI E.: *Metoda tabu w problemach szeregowania zadań produkcyjnych*. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 1999.

[10] PAWLAK M.: *Algorytmy ewolucyjne jako narzędzie harmonogramowania produkcji*. Wydawnictwo Naukowe PWN, Warszawa, 1999.

[11] RADELECZKI S., TÓTH T., GÖNDRI-NAGY J.: *A Multiple (Extended) Application of the Johnson Algorithm for the Two-Machine Manufacturing Cell Scheduling Based on Group Technology*. Production Systems and Information Engineering, Vol. 1, 2003, pp. 55–69.

[12] SKOŁUD B., WOSIK I.: *Algorytmy immunologiczne w szeregowaniu zadań produkcyjnych*. Zarządzanie Przedsiębiorstwem, Nr 1 (2008), s. 47–48.

[13] SOBASZEK Ł., GOLA A.: *Zastosowanie Matlaba w szeregowaniu zadań produkcyjnych*. [W:] Janczarek M., Lipski J., Technologie informacyjne w technice i kształceniu, Wyd. Politechniki Lubelskiej, Lublin 2013, s. 101-114. [14_moje metody]

[14] SOBASZEK Ł.: *Problemy harmonogramowania w systemach produkcyjnych*. Technological Complexes, nr 1, 2013, s. 175–178.

[15] SOTSKOV YU. N., SHAKHLEVICH N. V.: *NP-hardness of Shop-Scheduling Problems with Three Jobs*. Discrete Applied Mathematics, 59 (1995), pp. 237–266.

[16] SWOBODA L.: *Narzędzia planowania produkcji w zintegrowanych systemach informatycznych klasy ERP, ze szczególnym uwzględnieniem optymalnego harmonogramowania zagregowanych zleceń*. Innowacje w zarządzaniu i inżynierii produkcji – t. 1, [red:] Knosala R., Opole: Oficyna Wydawnicza Polskiego Towarzystwa Zarządzania Produkcją, 2014, s. 670–681.

[17] VILCOT G., BILLAUT J-CH.: *A Tabu Search and a Genetic Algorithm for Solving a Bicriteria General Job Shop Scheduling Problem*. European Journal of Operational Research, 190 (2008), pp. 398–411.

[18] YOUJUN CHENA, LINGFA LU, JINJIANG YUAN.: *Preemptive Scheduling on Identical Machines with Delivery Coordination to Minimize the Maximum Delivery Completion Time*. Theoretical Computer Science, Vol. 583, 2015, pp. 67–77.