

Scheduling for production teams

Yuri Mauergauz*

Sophus Group, Moscow 119296, Russian Federation

CHRONICLE

Article history:

Received October 14 2014
 Received in Revised Format
 February 10 2015
 Accepted March 17 2015
 Available online
 March 18 2015

Keywords:

Scheduling
Production intensity
Order utility

ABSTRACT

This paper presents a method of calendar (weekly) scheduling for production teams, when the average orders utility function is used as the quality criterion. The method is based on the concept of “production intensity”, which is a dynamic parameter of production process. Applied software package allows scheduling for medium quantity of jobs. The result of software application is the team load on the planning horizon. The computed schedule may be corrected and recalculated in interactive mode. Current load of every team is taken into account at each recalculation. The method may be used for any combination of complex and specialized teams.

© 2015 Growing Science Ltd. All rights reserved

1. Introduction

Contemporary methods of production planning consider each department of an enterprise, for which the schedule is developed, as the “working center”. This paper relates to scheduling for working centers, which are production teams. The production teams are organized for assembly of compound machines, equipment repair or transfer to a new place and so on. Each production team consists of regular or temporary staff, completing a joint job (order) and having joint responsibility for their results. Team members may have the same or different professions. Accordingly, there are specialized or complex teams. Each job may comprise some tasks (operations) of various types that require personnel of appropriate professions. Depending on compliance between task types and professions of team workers, a complex team may perform all tasks of a job or its part only. A specialized team may only perform a task corresponding to its profile.

The main advantage of using teams as working centers is associated with great flexibility inherent in such systems. Two types of flexibility are possible in scheduling. Routing flexibility is possibility to choose among two or more working centers to carry out a given operation. According to the classification (Blackburn & Millen, 1986) this type of flexibility is associated with hardware flexibility. The other type of flexibility named as sequencing flexibility is associated with software flexibility. This type of flexibility makes it possible to change the sequence of operations within the job. The possibility to

* Corresponding author.

E-mail: prizas@yandex.ru (Y. Mauergauz)

appoint one of available complex teams when scheduling for any job relates to routing flexibility of scheduling. At the same time, the sequence of operations needed for a job may be changed by agreement of team members, depending on such factors as worker's load, availability of facilities, etc.

Scheduling for teams is a difficult problem as it is often difficult for a manager to determine the load level of each worker and expected completion dates of the team tasks. The only possible version of production planning here is the combination of tasks scheduling for all teams within a planning period (a week, a month) and the daily plan, which is elaborated by the team itself. Since for completion of a specific planned job several teams of various specialization may be engaged, sequence of their work may only be directly determined by team leaders and may vary depending on the situation.

If production teams are considered as "machines", then according to the classification of planning problems (Graham et al., 1979), the set of production teams with non-determined sequence of their use may be determined as Open Shop. At the same time, possibility to several teams for a given operation provides the flexibility of such machine set. Therefore, we may suggest that scheduling for the set of production teams is associated with the Flexible Open Shop problem.

Apparently, the better is the preliminary schedule the less is the necessity of its revision and daily amendment. The schedule quality depends on the data set available to a planner at the moment of scheduling, and on the planning algorithm. In this article the criterion of average orders utility is used for scheduling estimation. The remainder of this paper is organized as follows. Section 2 provides a review of the relevant literature. Section 3 determines the main notions and the function of current order utility. In Section 4 the problem is formulated and the algorithm for its solving is provided. The example of scheduling for a given set of jobs is provided in Section 5. Section 6 contains some concluding remarks.

2. Literature review

As far as the author knows, there is only one article, which is dedicated to scheduling for technical structure of Flexible Open Shop type. In the paper by Witkowski et al. (2011) the problem of this type is considered, when all jobs at the moment of planning are available. In this paper the makespan C_{\max} is considered as a criterion. The criterion C_{\max} has been used in most other studies on Open Shop Scheduling problem. Gonzalez and Sahni (1976) elaborated the exact algorithm for this problem, when preemption is possible. Bai and Tang (2013) considered the task with the given release dates. In the book (Gupta et al., 2013) several algorithms were studied for scheduling at two operation stages with the criterion C_{\max} . Shabtai and Kaspi (2006) researched the Open Shop Scheduling problem with the criterion C_{\max} when job duration may vary. In some papers other criteria are used. Brasel et al. (2008) designed some heuristic algorithms for the Open Shop Scheduling problem with the criterion of mean flow time \bar{F} . Liaw (2005) studied this problem for the case of summary tardiness minimization $\sum T_i$, Naderi et al. (2011) used the criterion of summary duration minimization $\sum C_i$. Doulabi (2010) elaborated the algorithm for the Open Shop Scheduling problem based on the method of mixed integer linear programming. They used the criterion of earliness–tardiness penalties minimization.

The important version of the Open Shop Scheduling problem relates to a task of Concurrent Open Shop Scheduling. The latter may be considered as a variant of the classical Open Shop model, in which operations belonging to the same job may be processed concurrently on several machines. Ng et al. (2003) studied this problem for the criterion of weighted tardy jobs number $\sum w_i U_i$. Leung et al. (2005) used definition of the task as PD Open Shop Scheduling, if a certain machine is assigned to each operation. For example, Mastrolilli et al. (2010) for this problem applied the linear programming method

with the criterion of minimal summary weighted duration $\sum w_i C_i$; Ma and Jin (2013) also used linear programming with the criterion C_{\max} . All criteria above do not take into account the process time of planned jobs, which is a substantial disadvantage. For the Open Shop Scheduling problem Blazewicz et al. (2004) developed “non-classic” quality criterion, which is equal to the summary processing time of tardy jobs $Y = \sum Y_i$. In this criterion for each job $Y_i = \min(T_i, p_i)$, i.e. this value is equal to a smaller value of job tardiness T_i and processing time p_i .

3. Main problem definitions and utility functions

Let us assume that it is necessary to perform n various jobs at the facility of scheduling (enterprise, vessel, building, etc.) for a certain period of time after the planned start. The planning horizon of the working enterprise is usually equal to a certain reporting period, for example, a month or a week. When a vessel or a building is constructed, the horizon may be determined for the reporting period, or to the entire period of construction. Let us assume each job may be performed by one or several production teams, and their number is equal to M . Each job i comprises several tasks (operations) and has to be completed on due date d_i .

Assumptions

- Within this scheduling the sequence of tasks (operations) that belong to the same job to be performed was not taken into account. In general, these tasks may be executed in any sequence.
- The priority coefficient can be determined for each job.
- Release date is known for every job.
- Each job can be executed by any number of production teams simultaneously.
- Process time of a job as a whole and process time of every task belonging to a job is known and deterministic.
- The summary duration of job completion is assigned normatively.
- Each job has one task of a certain type as the main one.
- In the beginning, for every production team it is known what a job is being performed, and when this work will be completed.

Notation

Indices

- $i = 1, 2 \dots n$ Index of order (job)
 $l = 1, 2 \dots J$ Index of operation execution tree level
 $m = 1, 2 \dots M$ Index of specific production team in team list
 $j = 1, 2 \dots S$ Index of task (operation) type
 $z = 1, 2 \dots Z_l$ Index of decision tree node on level l

Parameters

- d_i Due date of job i in calendar days
 G Duration of plan period in calendar days
 E Quantity of hours per work day
 α “Psychological” coefficient
 h Horizon of planning in work days
 p_{ij} Processing time of operation j for job i in hours
 r_i Release moment for job i in calendar days

w_i	Weight coefficient for job i
p_i	Normative duration for job i in hours
N_i	Main operation type for job i

Variables

g_i	Necessary release moment of job i in calendar days
S_i	Pool of teams, which are able to perform the task of main type for job i
C_{lz}	Operation completion moment for tree node z at level l in hours
C_m	Last job completion moment for team m in hours
t	Current time
$t_{l+1,k}$	Start moment for operation k at level $l+1$ in hours
H_i	Current production intensity of job i
V_i	Current utility of job i
V	Current utility of all remaining operations
\bar{V}_l	Average utility of all jobs at level l
n_m	Number of jobs for team m
$\bar{V}_{l+1,k}$	Average utility of all jobs at level $l+1$ for operation k

Criterion

\bar{V} Average utility of all jobs on the planning horizon

Let us assume that the average job duration in calendar days is 1.4 more than the corresponding duration in working days. Accordingly, the necessary release date for job i is

$$g_i = d_i - 1.4 p_i / E. \quad (1)$$

The customer service level may be assessed with the help of the current orders utility function V . From the manufacturer's point of view, the order value increases proportionately to work amount p_i , since staff engagement increases. Besides, the more the time reserve is for completing an order, the more attractive is the order, since there is an opportunity to prepare for order execution. Eventually the order time reserve is decreasing, and the order value is diminishing. After all, if due date has expired, the order value becomes negative.

The manufacturer's attitude to the order changes with time, and the appropriate function is named *production intensity* (Mauergauz, 2012).

$$H_i = \frac{w_i p_i}{EG} \frac{1}{(d_i - t) / \alpha G + 1} \text{ at } d_i - t \geq 0$$

$$H_i = \frac{w_i p_i}{EG} [(t - d_i) / \alpha G + 1] \text{ at } d_i - t \leq 0 \quad (2)$$

The diagram of this function is shown in Fig. 1. X-axis in this figure represents time reserve. The reserve is equal to subtraction between due date and current time. In the positive part of the diagram ($d_i > t$) the values of intensity decrease in hyperbolic mode as available time reserve is growing. When the time

reserve is negative ($d_i < t$) and there is delay of order completion, the production intensity linearly increases. Since production intensity is dimensionless, it has no physical sense, but it has psychological sense. Indeed, when order delay is augmenting, nervousness about order execution is increasing. Two curves in Fig. 1 differ in the psychological coefficient value. The higher is the α coefficient, the more placid is the attitude to delays and the lower is the intensity.

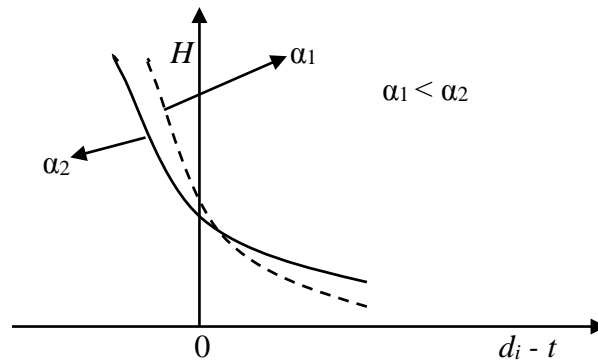


Fig. 1 Production intensity dependence on time reserve

The production intensity concept may be used for determination of the current order utility function V . The current utility (Fig. 2) for an order i is (Mauergauz, 2012):

$$V_i = \frac{w_i p_i}{EG} - H_i. \tag{3}$$

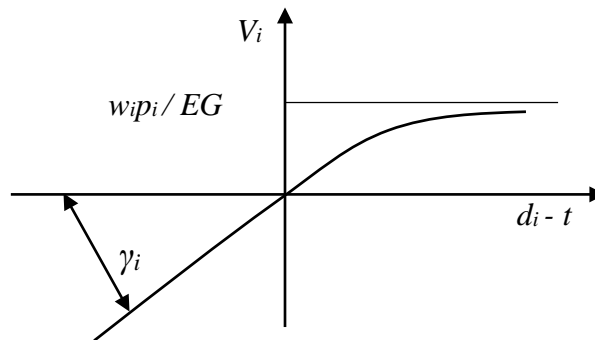


Fig. 2. Order utility function

The curve in Fig. 2 for the positive value $d_i - t \geq 0$ tends to the horizontal asymptote

$$V_i = w_i p_i / EG \tag{4}$$

In the negative part $d_i - t \leq 0$ the curve turns into the inclined straight line with

$$tg \gamma_i = \frac{w_i p_i}{\alpha EG^2}. \tag{5}$$

Assume the order quantity on the planning horizon equals n . Then their total utility V amounts to the sum of all order utilities, because orders, as a rule, are independent, i.e.

$$V = \sum_{i=1}^N V_i = \frac{1}{EG} \sum_{i=1}^N w_i p_i - \sum_{i=1}^N H_i. \tag{6}$$

The value of the function V changes in time, since the time reserve to the moment of scheduled execution changes. Besides, some orders are completed and new orders appear.

This paper suggests building a multi-level tree of works performance for making a schedule of parallel operating teams, while consistently considering feasibility of adding new tree nodes. Besides, these nodes determine both new jobs and teams that perform these jobs. New branches of the tree outgoing from each built node, may correspond both to the parent node team or other teams. Let us assume that a certain job corresponding to the tree node at the level l is completed at the moment of time C_{lz} . At the next level one of a set of uncompleted jobs has to be performed. Let us also assume that the job k starts at the moment t_{lk} , so this job has to be completed at the moment $t_k + p_k$. It was shown in (Mauergauz, 2013), the whole utility value at each level of the job fulfillment tree can be calculated with one of the recurrent formulas, which are used if utility value is known at the previous level:

$$\bar{V}_{l+1,k} = \frac{1}{t_k + p_k} (\bar{V}_l \times C_{zl} + \int_{C_{zl}}^{t_k + p_k} V_k dt), \quad \text{if } t_k + p_k \geq C_{zl} \quad (7)$$

$$\bar{V}_{l+1,k} = \bar{V}_l + \frac{1}{C_{zl}} \int_{C_{jq}}^{t_k + p_k} V_k dt, \quad \text{if } t_k + p_k < C_{zl}. \quad (8)$$

In Eqs. (7-8) the variable value \bar{V}_l is equal to average utility of the whole set of jobs, which are planned from the start $t = 0$ until completion C_{zl} of the last planned job; j is a team number, which is selected for a job J_k in a new tree branch; C_{jq} is a completion moment of the last job, which was performed by the team j at the level q ; V_k is the current utility of the whole set of all non-completed jobs; $\bar{V}_{l+1,k}$ is the average utility of the whole set of all planned jobs between the start moment $t = 0$ and the completion moment $t_k + p_k$ of the next planned job.

The upper integral limits in Eq. (7) and Eq. (8) are the same and equal to the completion moment of a job, which is planned at the next tree level construction. The bottom integral limits in Eq. (7) and Eq. (8) are different since they are equal to work completion moments of different teams. In the Eq. (7) this is the team z , which corresponds to the last built level; in the Eq. (8) this is the team j , which must perform a job at the next level. Apparently, this team may be used for tree construction at the previous levels, so the last level of its participation in scheduling $q < l$. The rules of integral calculations in Eqs. (7-8) are determined in (Mauergauz, 2014).

4. Problem formulation and its solution algorithm

In accordance with the well-known three-part scheduling classification, the problem may be recorded as:

$$FO | d_i, r_i | \bar{V}, \quad (9)$$

where FO is the designation of Flexible Open Shop problem, in which any sequence of parallel working teams is possible.

To solve the problem (9), we should apply a greedy algorithm, in which at each step a solution is selected with the best corresponding criterion. In problems with the average utility criterion \bar{V} there is an additional condition of start due date g_i (Mauergauz, 2014). Since the number of solution-selecting conditions, which are possible at each step, is equal to 2, the number of such solutions (tree nodes) may be more than 1. Accordingly, the solution tree comprises all branches that grow from each node. At each

step some possible solution versions that do not dominate each other are selected, and the algorithm below is used.

Step 1. (Initial computation of utility functions)

Let us assume that the level number is $l=0$; the initial orders utility function V_0 may be computed by the formula (6); number of nodes $Z_0=1$.

External cycle

Step 2. (Determination of possible operations at next levels)

For each node z of the built tree at the level l all possible jobs k at the level $l+1$ are determined, and values g_k are computed by the formula (1).

Intermediate cycle

Step 3. (Determination of necessary main teams at next levels)

For each possible job the pool S_i of teams, which are able to perform the main task of this job, is determined.

Step 4. (Determination of necessary subsidiary teams at next levels)

For each possible job the least charged teams, which are not the main for this job, are determined.

Internal cycle

Step 5. (Utility function computation at next levels)

For each team m related to the pool S_i values $\bar{V}_{l+1,k,m}$ are computed using the formulas (7,8).

End of internal cycle

End of intermediate cycle

Step 6. (Determination of dominated tree nodes)

If the level $l+1$ is not last, then for domination on the level $l+1$ of the tree node y with a job i over the tree node x it is sufficient to comply with the following equations

$$\bar{V}_{l+1,y} \geq \bar{V}_{l+1,x} \text{ and } g_{l+1,y} < g_{l+1,x}, \quad (10)$$

Otherwise: on the last level $l+1$ domination is possible, if

$$\bar{V}_{l+1,y} \geq \bar{V}_{l+1,x} \quad (11)$$

Step 7. (Transition to the next level or stopping)

If the level is more than the last (all operations are completed), then STOP.

Otherwise: level number increment $l = l+1$ and go to Step 2.

End of external cycle

5. Example of teams load scheduling with given set of jobs

Let us assume it is necessary to perform 40 various jobs at the facility of scheduling (enterprise, vessel, building etc.) during some period after planned start. For example, assume there is maintenance of various equipment going on, and each job comprises tasks, every of which relates to one of five possible types (Table 1).

Table 1
Types of tasks (operations)

Type No.	Type name
1	Mechanical
2	Electrical
3	Instrument
4	Building
5	Transport

Assume in this case there are five teams, which may be drawn in. Besides, some of them are specialized and can perform tasks of a single type, and other teams are complex. In Table 2 the example of highest throughput for each task type is shown.

Table 2
Highest team throughput in hours/day

Task type No.	Team No.				
	1	2	3	4	5
1	24	16	16	0	0
2	8	16	8	0	0
3	0	8	8	0	0
4	0	0	0	16	0
5	0	0	0	0	16

Each planned job is described by a whole set of parameters. In Table 3 the fragment of the main set of data on jobs is shown.

Table 3
Fragment of main data on jobs

Job No.	Calendar due date	First possible day	Whole process time in hours	Normative duration in working days	Main task type	Weight coefficient	Completion mark
1	-2	0	50	4	1	5	0.5
2	3	0	200	10	1	1	0.5
3	5	0	50	4	5	1	0.5
4	5	0	150	10	1	1	0.5
5	7	0	100	7	2	1	0.5
6	9	0	50	4	5	1	0
7	9	0	150	10	3	1	0
8	12	8	100	7	1	1	0
9	12	8	50	4	2	1	0
10	14	8	150	10	2	1	0

From Table 3 it follows that jobs are sorted by increment of due date, and this date on a day of scheduling may be exceeded. The job due date, the possible start date, the whole process time and the job duration are defined by given norms for scheduled maintenance. Any job has priority, which is determined by the weight coefficient. If a job has already been started at the moment of scheduling, the value 0.5 is input into the column "Completion mark". Since, as it was mentioned above, a job may consist of five tasks, Table 3 specifies the main task for each job, i.e. the one that determines possibility to start the entire job. For scheduling it is necessary to have initial data for started jobs at the planning moment (Table 4). If a task has to be completed but its execution has not started, the number -1 is input into the appropriate column.

Table 4
Fulfillment of started jobs at planning moment in %

Job No.	Task type	Team No.				
		1	2	3	4	5
1	1	0	80	0	0	0
1	2	0	-1	0	0	0
2	1	20	0	0	0	0
2	2	20	0	0	0	0
2	3	0	-1	0	0	0
3	4	0	0	0	-1	0
3	5	0	0	0	0	80
4	1	0	0	70	0	0
4	2	0	0	100	0	0
4	3	0	0	-1	0	0
4	4	0	0	0	100	0
5	1	0	30	0	0	0
5	2	0	70	0	0	0
5	5	0	0	0	0	60

As it follows from Table 4, at the moment of scheduling the team 1 performs only the job 2, and the team 3 performs the job 4. The team 2 at this moment started execution of the jobs 1 and 5; besides, it must take part in the job 2. The team 4 completed its part of the job 4, and must take part in the job 3. The team 5 performs the jobs 3 and 5 in parallel. In scheduling process it is necessary to take into account that the team throughput during following weeks may be below high values in Table 2, because some workers may be absent for any reasons. A special table for expected throughput on the planning horizon of 6 weeks is made. All tables are recorded in the MS Excel sheet.

To search for solutions on the basis of the criterion \bar{V} , the program using the VBA language for MS Excel has been designed. In Fig. 3 the results of scheduling computation are shown, when parameters have the following values: psychological coefficient $\alpha=0.1$; duration of plan period $G=5$ days; quantity of hours per work day $E=8$. For each job calendar days of the start and the completion are recorded. Numbers in brackets correspond to the teams, which take part in the job; the team, which executes the main job component, is the first.

	A	B
1	Job 1: 0 – 2 (2)	
2	Job 2: 0 – 12 (1, 2)	
3	Job 3: 0 – 2 (5, 4)	
4	Job 4: 0 – 5 (3)	
5	Job 5: 0 – 5 (2, 5)	
6	Job 6: 0 – 6 (5, 3)	
7	Job 7: 0 – 14 (2, 4)	
8	Job 8: 0 – 10 (1, 3)	
9	Job 9: 8 – 14 (2)	
10	Job 10: 8 – 22 (1)	
11	Job 11: 8 – 18 (2, 5)	
12	Job 12: 8 – 18 (2, 4, 5)	

Fig. 3. Fragment of jobs distribution

Planned teams load for the following seven weeks is shown in Fig. 4. In this case we can see that during the first and the second weeks the team load is not more than 100%; in the third week all teams are overloaded, in following weeks there are load oscillations. The computed plan solution generally ensures timeliness of jobs completion, but it is not optimal, and may be substantially improved. For this purpose,

it is necessary to analyze the list of jobs, which are planned for the teams on the planning horizon. In Fig. 5 the fragment of the team timetable for jobs execution in the following 3 weeks is shown. Information in Fig. 4 shows that the team 3 is the most overloaded team, which on the third week has the load of 161%. Studying the list of jobs for the team 3 in the third week, we can suggest that this load could be less, if execution of the job 22 was postponed to the next week.

	A	B	C	D	E	F	G	H
13								
14	Weekly teams load in % of throughput							
15	Weeks	1	2	3	4	5	6	7
16	Team 1:	72	83	112	62	146	136	28
17	Team 2:	76	86	100	112	121	49	0
18	Team 3:	96	76	161	65	125	76	9
19	Team 4:	52	95	128	58	142	85	16
20	Team 5:	50	28	112	78	138	98	10

Fig. 4. Planned teams load

	A	B	C	D	E	F	G	H
21								
22	Weekly jobs list							
23	Weeks	1		2			3	
24	Team 1:	2, 8		2, 8, 10			10, 26, 20	
25	Team 2:	1, 2, 7, 5		2, 7, 11, 12, 9			11, 12, 20, 24, 30	
26	Team 3:	4, 8, 6		8, 14, 19, 17, 13, 15			14, 19, 17, 13, 15, 22	
27	Team 4:	7, 3		7, 12, 19, 13, 15			12, 19, 13, 15, 18, 24	
28	Team 5:	3, 5, 6		11, 12, 14			11, 12, 14, 26, 18, 30, 22	

Fig. 5. Fragment of jobs list scheduled for teams

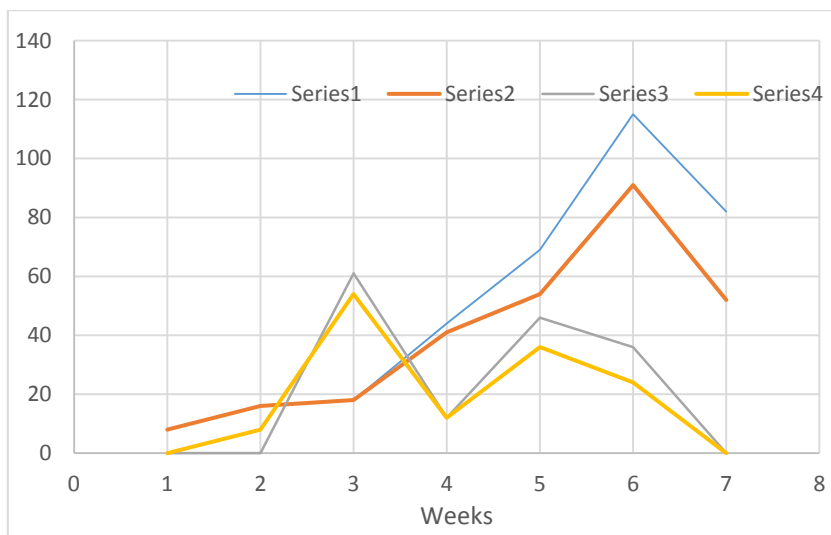


Fig. 6. Jobs tardiness and highest team overload

The plan improvement achieved by correction is shown on the diagrams in Fig. 6. The diagrams show the summary tardiness for all planned jobs in hours, and the overload of the most charged teams as percentage on the planning horizon. Thin lines refer to the initial planning solution, thick lines relate to the corrected solution. As it follows from Fig. 6, the correction diminishes both the jobs tardiness and

the most team overload, appreciably. However, it is impossible to eliminate unevenness completely using only the scheduling process. Therefore, for timely jobs completion we have to change the team structure or duration of working time.

- 1) Summary weekly jobs tardiness according initial schedule in hours.
- 2) Summary weekly jobs tardiness according corrected schedule in hours.
- 3) Overload of the charged team according to initial schedule as percentage.
- 4) Overload of the charged team according to corrected schedule as percentage.

6. Conclusion

Results above confirm that the approach to production teams scheduling, which is based on application of order utility functions, produces a satisfactory schedule. The problem solution, which uses the “greedy” algorithm, is able to construct the search tree starting from the initial system state. This algorithm determines the domination bound for tree nodes at each level according to the criterion \bar{V} , and in addition according to the value of necessary job release date g . The program in VBA language for MS Excel was designed, and the example of its application for scheduling was made. The calculated schedule may be improved in interactive mode, if a certain job is postponed to the next week.

Scheduling is a regular process that repeats with certain, but not always constant frequency. For this purpose it is convenient to use new MS Excel sheets, where information from previous sheets may be contained. By changing or inserting new data, the user may correct the previous plan or design a new one. Analyzing a set of calculated schedules, one may detect systematic deviations of teams load and optimize professional team structures. In practice various additional constraints may be necessary for scheduling. For example, often it is needed to take into account impossibility of job execution in a certain time interval. In the nearest future it is planned to elaborate some solutions for such problems.

References

- Bai, D., & Tang, L. (2013). Open shop scheduling problem to minimize makespan with release dates. *Applied Mathematical Modeling*, 37, 2008-2015.
- Blackburn, J., & Millen, R. (1986). Perspectives on flexibility in manufacturing: hardware versus software. *Modeling and DeMgn of Flexible Manufacturing Systems*, 2(2), 116-117.
- Błażewicz, J., Pesch, E., Sterna, M., & Werner, F. (2004). Open shop scheduling problems with late work criteria. *Discrete Applied Mathematics*, 134(1), 1-24.
- Bräsel, H., Herms, A., Mörig, M., Tautenhahn, T., Tusch, J., & Werner, F. (2008). Heuristic constructive algorithms for open shop scheduling to minimize mean flow time. *European Journal of Operational Research*, 189(3), 856-870.
- Doulabi, S. H. H. (2010). A mixed integer linear formulation for the open shop earliness-tardiness scheduling problem. *Appl Math Sci*, 4(35), 1703-1710.
- Gonzalez, T., & Sahni, S. (1976). Open shop scheduling to minimize finish time. *Journal of the ACM (JACM)*, 23(4), 665-679.
- Graham, R. L., Lawler, E. L., Lenstra, J. K., & Kan, A. R. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of discrete mathematics*, 5, 287-326.
- Gupta, D., Renuka, M., & Singla, P. (2013). *Open Shop Scheduling: Minimizing Makespan*. Saarbrücken, LAP Lambert.
- Leung, J. Y. T., Li, H., & Pinedo, M. (2005). Order scheduling in an environment with dedicated resources in parallel. *Journal of Scheduling*, 8(5), 355-386.
- Liaw, C. F. (2005). Scheduling preemptive open shops to minimize total tardiness. *European Journal of Operational Research*, 162(1), 173-183.

- Mastrolilli, M., Queyranne, M., Schulz, A. S., Svensson, O., & Uhan, N. A. (2010). Minimizing the sum of weighted completion times in a concurrent open shop. *Operations Research Letters*, 38(5), 390-395.
- YanMin Ma, Y. M., & Di Jin, D. J. (2013). Concurrent Open-shop Scheduling Accurate Algorithm Research. *International Journal of u-and e-Service, Science and Technology*, 6(5), 1-16.
- Mauergauz, Y. (2012). Objectives and constraints in advanced planning problems with regard to scale of production output and plan hierarchical level. *International Journal of Industrial and Systems Engineering*, 12(4), 369-393.
- Mauergauz, Y. (2013). Cost-efficiency method for production scheduling. In *Proceedings of the world congress on engineering* (Vol. 1, pp. 587-593).
- Mauergauz, Y. (2014). Dynamic Pareto-optimal group scheduling for single machine. *International Journal of Industrial and Systems Engineering*, 16, 537-559.
- Naderi, B., Ghomi, S. F., Aminnayeri, M., & Zandieh, M. (2011). Scheduling open shops with parallel machines to minimize total completion time. *Journal of Computational and Applied Mathematics*, 235(5), 1275-1287.
- Ng, C. T., Cheng, T. C. E., & Yuan, J. J. (2003). Concurrent open shop scheduling to minimize the weighted number of tardy jobs. *Journal of Scheduling*, 6(4), 405-412.
- Shabtay, D., & Kaspi, M. (2006). Minimizing the makespan in open-shop scheduling problems with a convex resource consumption function. *Naval Research Logistics (NRL)*, 53(3), 204-216.
- Witkowski, T., Antczak, P., & Antczak, A. (2012, February). Hybrid method for solving flexible open shop scheduling problem with simulated annealing algorithm and multi-agent approach. In *Advanced Materials Research* (Vol. 383, pp. 4612-4619).