www.czasopisma.pan.pl PAN www.journals.pan.pl

ARCHIVES of FOUNDRY ENGINEERING

DOI: 10.1515/afe-2017-0105

Published quarterly as the organ of the Foundry Commission of the Polish Academy of Sciences

DE GRUYTER OPEN

ISSN (2299-2944)
Volume 17
Issue 3/2017

133 – 138

Coordinated Production Planning and Scheduling Problem in a Foundry

A. Stawowy, J. Duda *
AGH University of Science and Technology, Faculty of Management
ul. Gramatyka 10, 30-067 Kraków, Poland
*Corresponding author. E-mail address: jduda@zarz.agh.edu.pl

Received 30.03.2017; accepted in revised form 05.07.2017

Abstract

In the paper, we present a coordinated production planning and scheduling problem for three major shops in a typical alloy casting foundry, i.e. a melting shop, molding shop with automatic line and a core shop. The castings, prepared from different metal, have different weight and different number of cores. Although core preparation does not required as strict coordination with molding plan as metal preparation in furnaces, some cores may have limited shelf life, depending on the material used, or at least it is usually not the best organizational practice to prepare them long in advance. Core shop have limited capacity, so the cores for castings that require multiple cores should be prepared earlier. We present a mixed integer programming model for the coordinated production planning and scheduling problem of the shops. Then we propose a simple Lagrangian relaxation heuristic and evolutionary based heuristic to solve the coordinated problem. The applicability of the proposed solution in industrial practice is verified on large instances of the problem with the data simulating actual production parameters in one of the medium size foundry.

Keywords: Heuristics, Application of information technology to the foundry industry, Mixed integer programming, Production planning, Scheduling

1. Introduction

Increasing competition and globalization causes that companies increase the range of produced castings, shorten the time needed to prepare and implement their technology into production. Therefore, in modern foundries flexible systems consisting of mechanized and automated equipment are used, ensuring fast, efficient and good quality production of castings, usually on a small scale.

In this paper we studied an integrated two-level lot-sizing and scheduling problem with assembly stage in a mid-size foundry that produce alloy castings in a make-to-order system, which allows consumers to purchase products that are fully customized to their specifications. In such case, we are dealing with the instance of a problem belonging to Capacitated Two-Level Lot Sizing and Scheduling Problem (CTLSP) [3][5]. At the first level the raw materials are prepared and the decisions about quantity of produced cores and schedule for melting alloys must be made. At the same time, the lot size of ordered items and its corresponding production schedule for the molding line must be determined in the second level. We assume that the finite planning horizon is subdivided into smaller sub-periods (determined by furnace loads), and melting furnace and cores shop have limited capacity.

The aim of presented paper is to develop the model and appropriate way of its solution able to achieve optimized production plans for core shop, melting furnace and molding line within few minutes, which is usually enough for the planners in real production conditions (plans are prepared usually at the beginning of each working day, but sometimes need to be rearranged, if some unexpected even occurs like order cancelation or machine breakdown). More complete review of the papers dealing with production planning in foundries can be found e.g. in [6].

The coordination problem of the furnace and molding line schedules with the schedules for the other shops in a foundry has so far not been investigated extensively in the literature. Hans and de Velde [4] in their mixed integer programing model checked cores availability and the cooling space capacity. Since CPLEX solver was not able to solve such a complex model, they proposed a hierarchical approach in which they combined mixed integer linear programming, shortest path algorithms, and iterative local improvement. According to the authors their solution was much better than manual approach used in the analyzed foundry, however, only relatively small case with 17 items and 20 periods was analyzed in details.

Our goal is to provide the solution for the industrial size combined lot-sizing and scheduling problem of molding line and the core shop producing cores that must be assembled before the melted alloy is poured into the molds.

The paper is organized as follows: Definitions of the problem and notation are described in Section 2. Section 3 presents the details on proposed heuristic approaches. The computational experiments are summarized in Section 4, and the conclusions are drawn in Section 5.

2. Lot-sizing and scheduling model

In this section we formulate the mixed integer programming model to the problem of alloy casting production with cores assembly. In its castings production planning part it is similar to the model presented by Araujo et al. proposed for an automated foundry [1]. It contains two decision variables and four constraints. The second part of the model reflecting the production of cores that should be assembled before alloy castings production contains additional decision variable and two constraints. We use the following notation:

Indices

$i = 1,\dots, I$ produced items (alloy castings);
$k = 1,\dots, K$ produced alloys (metal grades);
$t = 1,\dots, T$ working days;
$n = 1,\dots, N$ sub-periods in day t;
$\tau = -2,\dots, T$; days in which cores are be produced (at most two day before the assembly);
$\eta = 1,\dots, N$; sub-periods in day τ.

Parameters

d_{it} demand for item i in day t;
w_i weight of item i;
c_i number of cores that should be assembled for item i;
a_i^k 1, if item i is produced from alloy k, otherwise 0;
st setup cost for alloy change in furnace;
C loading capacity of the furnace;
CC_{tn} cores production capacity in day t and period n;
dc_i penalty for delaying production of item i;
hc_i holding costs of item i;
I_{i0}^- initial number of items i delayed at the end of day t;
I_{i0}^+ initial inventory of items i at the end of day t;

Variables

I_{it}^- number of items i delayed at the end of day t;
I_{it}^+ inventory of items i at the end of day t;
z_{tn}^k 1, if there is a setup (change) of alloy k in sub-period n of day t, otherwise 0;
y_{tn}^k 1, if alloy k is produced in n in sub-period of day t, otherwise 0;
x_{itn} number of items i produced in sub-period n of day t;
$xc_{i\tau\eta tn}$ number of cores produced in sub-period η of day τ for item i in sub-period t of day n.

Production planning problem with cores assembly is then defined as follows:

$$\text{Min} \quad \sum_{i=1}^{I}\sum_{t=1}^{T}(dc_{it}I_{it}^- + hc_{it}I_{it}^+) + \sum_{k=1}^{K}\sum_{n=1}^{N}(st\ z_{tn}^k) \tag{1}$$

subject to:

$$I_{i(t-1)}^+ - I_{i(t-1)}^- + \sum_{n=1}^{N}\sum_{k=1}^{K} x_{itn}a_i^k - I_{it}^+ + I_{it}^- \ge d_{it} \qquad i = 1,\dots,I,\ t = 1,\dots,T \tag{2}$$

$$\sum_{i=1}^{I} w_i x_{itn} a_i^k \le C y_{tn}^k \qquad k = 1,\dots,K,\ t = 1,\dots,T, n = 1,\dots,N \tag{3}$$

$$z_{tn}^k \ge y_{tn}^k - y_{t(n-1)}^k \qquad k = 1,\dots,K,\ t = 1,\dots,T, n = 1,\dots,N \tag{4}$$

$$\sum_{k=1}^{K} y_{tn}^k = 1 \qquad t = 1,\dots,T, n = 1,\dots,N \tag{5}$$

$$\sum_{i=1}^{I}\sum_{t=1}^{T}\sum_{n=1}^{N} xc_{i\tau\eta tn} \le CC_{\tau\eta} \qquad \tau = -2,\dots,T, \eta = 1,\dots,N \tag{6}$$

$$\sum_{k=1}^{K} x_{itn}a_i^k c_i \le \sum_{\tau=-2}^{T}\sum_{\eta=1}^{N} xc_{i\tau\eta tn} \qquad i = 1,\dots,I, t = 1,\dots,T, n = 1,\dots,N \tag{7}$$

$$\begin{gathered} I_{it}^-, I_{it}^+, x_{itn}, xc_{i\tau\eta tn} \ge 0 \\ i = 1,\dots,I, t = 1,\dots T, n = 1,\dots,N, \tau = -2,\dots,T, \eta = 1,\dots N \\ z_{tn}^k, y_{tn}^k \\ k = 1,\dots,K, t = 1,\dots T, n = 1,\dots,N \end{gathered} \tag{8}$$

The goal (1) of the coordinated planning is to find the plan that minimizes the total sum of the costs of delayed production, storage costs of finished goods and the setup cost resulting from alloy change during furnace load.

Constraint (2) balances inventories, delays and the volume of production of each item in a particular day. The production of alloys in a sub-period is summarized for that day and compared with daily demand. Constraint (3) ensures that the furnace capacity is not exceeded in a single load by summarizing all items made out of the same alloy grade. Constraint (4) sets variable z_n^k to 1, if there is a change in an alloy grade in the subsequent sub-periods. Constraint (5) ensures that only one alloy is produced in each sub-period. Constraint (6) ensures that the capacity of the core shop is not exceeded in a given sub-period. Constraint (7) checks if there is enough cores to be assembled in the mold in order to produce the desired number of items. We assume that cores can be prepared not earlier than two days before using in the mold. In order to control the time two groups of indices are used – one for the production time of a core (τ and η) and second indicating for which particular day and sub-period it will be used (t and n). Finally all the variables should be integers greater than 0, except for variables z_{tn}^k and y_{tn}^k - indicating the changes of alloy that are binary

3. Solution methods

Initial experiments indicated that the presented MIP formulation of the problem was very hard to solve for CPLEX solver, even for the smallest instances considered (with 50 items and 10 alloys), so we proposed some heuristics that significantly improved the initial results. First approach is based on the Lagrangian relaxation of the proposed model, the second one uses relax and fix strategy, while the last one is based on the evolutionary approach.

3.1. Simple relaxation of the problem

Introducing a new decision variable responsible for cores production planning significantly complicated the problem (for 50 items, 5 days and 10 periods we have 175,000 integer variables, and twice as much for 100 items). We then decided to provide simple Lagrangian relaxation for constraint (7).
Additional variables are used and constraint (7) it is calculated as follows:

$$xc_{dev} = \max(0, \sum_{i=1}^{I}\sum_{t=1}^{T}\sum_{n=1}^{N}(\sum_{\tau=-2}^{T}\sum_{\eta=1}^{N} xc_{i\tau\eta tn} - \sum_{k=1}^{K} x_{itn} a_i^k c_i)) \tag{9}$$

A relaxed objective function (1) has the following form:

$$\text{Min} \quad \sum_{i=1}^{I}\sum_{t=1}^{T}(dc_{it} I_{it}^{-} + hc_{it} I_{it}^{+}) + \sum_{k=1}^{K}\sum_{n=1}^{N}(st\, z_{tn}^{k}) + \lambda\, xc_{dev} \tag{10}$$

Such relaxation brings significant improvement of the final results, what will be shown later in the experiments.

3.2. Rolling horizon relaxation

In order to further improve the results we adapted a relax and fix heuristic proposed by Araujo et al. [1] for the lot-sizing problem in a foundry. They relaxed variables x_{in} and y_n^k for not fixed sub-periods. Variable x'_{in} representing the number of items i produced in sub-period n was turned into float type instead of integer, while y'^k_t was replaced by integer value reflecting for how many sub-periods given alloy grade will be produced.
Thus the constraint (2) balancing inventory and demand looks as follows:

$$I_{i(t-1)}^{+} - I_{i(t-1)}^{-} + \sum_{n=1}^{N}\sum_{k=1}^{K} x_{itn} a_i^k - I_{it}^{+} + I_{it}^{-} \ge d_{it} \qquad i = 1,\dots,I,\ t = 1,\dots,T \tag{11}$$

Similarly the constraint (3) is valid only for the fixed days (t_f) and for other days it looks as follows:

$$\sum_{i=1}^{I}\sum_{n=1}^{N} w_i x'_{in}\, a_i^k \le C y_t^k, \qquad k = 1,\dots,K,\ t = 1,\dots,T,\ t \ne t_f \tag{12}$$

Analogously the constraint (5) for the days other than the fixed one is extended to the following formula:

$$\sum_{k=1}^{K} y'^k_t = N \qquad t = 1,\dots,T, t \ne t_f \tag{13}$$

For the assembly problem we have to deal not only with items (alloy castings), but also with cores for them. We propose an analogous relaxation as for the items. For the fixed day amounts of cores planned for this day must be also fixed, as we have to provide enough cores to be assembled in the molds for this particular day. For the remaining days variables can be relaxed using floating point variables $xc'_{i\tau\eta tn}$. As a result constraint (6) is replaced by the following constraint:

$$\sum_{i=1}^{I}\sum_{t=t_f}^{T}\sum_{n=1}^{N} xc_{i\tau\eta tn} + \sum_{i=1}^{I}\sum_{t \ne t_f}^{T}\sum_{n=1}^{N} xc'_{i\tau\eta tn} \le CC_{\tau\eta} \qquad \tau = -2,\dots,T, \eta = 1,\dots,N \tag{14}$$

And the constraint (7) is replaced by the two constraints:

$$\sum_{k=1}^{K} x_{itn} a_i^k c_i \le \sum_{\tau=-2}^{T}\sum_{\eta=1}^{N} xc_{i\tau\eta tn} \qquad i = 1,\dots,I, t = t_f, n = 1,\dots,N \tag{15}$$

and

$$\sum_{k=1}^{K} x'_{itn} a_i^k c_i \le \sum_{\tau=-2}^{T} \sum_{\eta=1}^{N} xc'_{i\tau\eta tn} \qquad (16)$$
$$i = 1,...,I, t \ne t_f, n = 1,...,N$$

However, those two constraints were relaxed using Lagrangian relaxation proposed in the previous point. The part of the objective function (10) responsible for penalizing not fulfilled constraints (15-16) looks as follows:

$$xc_{dev} = \max(0, \sum_{i=1}^{I} \sum_{t=t_f}^{T} \sum_{n=1}^{N} (\sum_{\tau=-2}^{T} \sum_{\eta=1}^{N} xc_{i\tau\eta tn} - \sum_{k=1}^{K} x_{itn} a_i^k c_i) + \sum_{i=1}^{I} \sum_{t\ne t_f}^{T} \sum_{n=1}^{N} (\sum_{\tau=-2}^{T} \sum_{\eta=1}^{N} xc_{i\tau\eta tn} + \sum_{\tau=-2}^{T} \sum_{\eta=1}^{N} xc'_{i\tau\eta tn} - \sum_{k=1}^{K} x'_{itn} a_i^k c_i) \qquad (17)$$

For the fixed day production of all cores necessary to be put in the mold must be strictly planned, while for the items in the remaining days we compare strict and relaxed production plans for cores with the relaxed production of items.

The model for rolling-horizon was computed T times. Each time values of fixed variables computed for the previous days were included as the constants for the current fixed day.

In order to automate the computing process we wrote a C# application that executes CPLEX to solve the rolling horizon models iteratively day after day, taking the values of the fixed variables from the previous solution. We set the time limit for each model to 2 minutes for problems with 50 items and 3 minutes for the problems with 100 items.

3.3. Evolutionary based heuristic (EBH)

We propose simple evolutionary based heuristic which was successfully applied to grouping problems [7], adapted to the CTLSP. It is variant of (1,λ)-Evolution Strategy (ES) where λ number of children is generated from one parent by means of the simple mutations; the crossover is not employed. The best of the descendants becomes the new parent solution (deterministic selection). We don't use a destabilization procedure which role is to shift the search to a new area in the solution space, as during algorithm runs we have observed continuous improvement of the results.

The representation of solution consists of two chromosomes: vector **A** of integers representing alloys' ID that are prepared in the coming sub-periods, and table **X** of integers representing the quantity of items that are produced. The vector **A** consists of $T*N$ elements, and the table **X** has the dimension of I rows by $T*N$ columns.

As the fitness function (FF) we use goal function for relaxed problem which is expressed in (10). After some preliminary experiments, the population size λ was decided to keep at the constant level equal to 40 descendants. The stopping criterion is maximum number of generations: the algorithm is terminated after 3,000,000 children have been generated and evaluated (i.e. after 75,000 generations).

Initial solution

The starting solution is generated at random but taking into account the data characterizing the problem. First, a demand for the various alloys is calculated and vector **A** is filled according to obtained probability distribution. Then, quantities x_{itn} of produced items are generated from a discrete uniform distribution within [0, d_{it}]. If sum_{tn} of castings' weights in any subperiod exceeds the furnace capacity C, the quantities are decreased proportionally to factor C/sum_{tn} during the evaluation of starting solution. Generating data and filling vector **A** together with table **X** are repeated 1000 times and the best from resulting individuals becomes the starting solution. The overall procedure creates feasible individual of reasonable quality.

The mutations

Two essential assumptions make the proposed algorithm powerful:

1. variety of problem-specific mutations working with time, quantities and alloys,
2. decreasing of mutation intensity during algorithm run.

We employed six mutation operators:

Mut1. choose at random one of the sub-periods and change its alloy to another drawn from K available.

Mut2. choose at random two different sub-periods and swap between them the quantities of manufactured products.

Mut3. like Mut2 but also sub-periods' alloys are swapped.

Mut4. choose at random tn sub-period and i product, and draw its quantity from normal distribution $N(x_{itn}, \delta)$, where x_{itn} is quantity of item i taken from previous generation (iteration), δ is parameter (standard deviation) responsible for the mutation intensity.

Mut5. choose at random tn sub-period and change the manufactured quantities for all I items using approach described in Mut4, regardless of produced in tn alloy.

Mut6. choose at random one of I items and change its manufactured quantities in all TN sub-periods using approach described in Mut4, regardless of produced alloys.

We can observe that the best mutation is Mut4 so we decided to set its probability to 0.25, while the other mutations have the same application probabilities equal to 0.15.

In canonical version of the ES individual consists of two chromosomes: each main element in solution has corresponding standard deviation parameter, and the mutation intensity (i.e. standard deviation) is subject to a self-adaptation process [2]. We could not use this approach due to a large dimension of **X** table which significantly would extend the time of calculations. We decided to set the same value δ for mutations Mut4-Mut6 and all individuals in the current generation. Moreover, it is assuming that the mutations intensity should decrease with the number of the current iteration, thus maintaining the exploration of search space in the initial phase of the algorithm and proper neighborhood exploitation of good solution in the final phase of the algorithm. After some preliminary experiments, we have chosen linear decreasing of mutations intensity from $\delta = 10.0$ in first generation to δ=0.5 in the last generation. It turned out that this approach works very well.

4. Computational experiments

4.1. Test problems

Computational experiments were conducted on the basis of the test problems prepared by the authors, but the parameters for castings were drawn according to the procedure described by Araujo et al. [A]. Two sizes of planning problems were considered: 50 items made from 10 different alloys and 100 items made from 20 alloys. The characteristic of these problems is presented in Table 1. The values for demand, weight and delaying cost were determined using uniform distribution within a given range.

Table 1.
Test problems characteristics

Parameter	**Value**
number of items (I), number of alloys (K)	(50,10); (100,20)
number of days (T)	5
number of subperiods (N)	10
demand (d_{it})	[10, 60]
weight of item (w_i)	[2, 50]
number of cores (c_i)	[0, 7]
setup penalty (st)	[15, 50]
delaying cost (h_i^-)	[3.00, 9.00]
holding cost (h_i^+)	w_i * 0.02 + 0.05

Ten instances of the problem for each size were generated. The basis furnace capacity C was obtained using the following formula corresponding to the total sum of the weights of ordered items:

$$C = \frac{\sum_{i=1}^{I}\sum_{t=1}^{T} d_{it} w_i + \sum_{k=1}^{K} st}{T * N} \tag{18}$$

The basis capacity CC of the core shop was determined in a similar way:

$$CC = \frac{\sum_{i=1}^{I}\sum_{t=1}^{T} d_{it} c_i}{T * N} \tag{19}$$

4.2. Results

We tested the proposed heuristic methods and compared them with the solution provided by the sate-of-the-art mixed integer programming (MIP) CPLEX solver (included in IBM OPL Studio 12.7). For the CPLEX solver time limit was set to 10 minutes for the instances with 50 items and10 alloys, and 15 minutes for the instances with 100 items and 20 alloys.

The results achieved for all 10 instances with 50 items are presented in Table 2, while the results for 100 items are shown in Table 3. Solution in MIP column provides the results for CPLEX solver using the model without any relaxation. Column LR shows the results for the Lagrangian relaxation. Column RH provides the results for the rolling horizon relax and fix approach, and EBH represents the results of evolutionary based heuristic in the following order: the best result out of ten runs, average result and the standard deviation of the solutions.

The proposed evolutionary heuristic performed on average significantly better than CPLEX without any relaxation of the model. CPLEX with relaxed versions of the model were better than EBH only by 4-7% when the best results are compared, and by 7-10% comparing the average results. The relaxed versions of the model differ by only 3%, among which the rolling horizon approach provide the best results.

Table 2.
Results of the experiments – problem (50, 10)

#	MIP	LR	RH	EBH_{best}	EBH_{avg}	EBH_{sd}
1	5816.54	5363.72	5239.04	5487.57	5851.86	299.72
2	5316.33	5002.64	5001.98	5154.77	5232.22	63.07
3	4091.32	3972.85	3956.99	4450.52	4560.00	79.33
4	6422.74	4192.42	4166.82	4740.62	4993.36	172.05
5	5859.56	5217.60	5149.94	5545.17	5742.71	104.13
6	4904.97	4670.89	4663.27	4906.99	5056.29	93.59
7	7032.92	5843.36	5771.59	6191.83	6247.44	60.53
8	5260.88	4892.18	4891.96	5018.58	5070.82	31.62
9	6989.56	6319.71	6315.04	6635.91	6767.64	95.80
10	4211.17	3719.66	3709.30	3913.29	4104.90	194.16
Avg.	5905.99	4919.50	4886.59	5204.53	5362.72	
St.d.	982.75	776.80	764.77	802.52	770.67	

Table 3.
Results of the experiments – problem (100, 20)

#	MIP	LR	RH	EBH_{best}	EBH_{avg}	EBH_{sd}
1	42013.4	25974.8	23631.8	26578.2	27110.4	349.8
2	47214.7	27312.6	26625.6	27676.8	28049.0	219.2
3	46775.6	23907.2	23225.9	23700.0	24193.4	405.5
4	48446.6	26223.0	25710.0	26525.0	26975.9	469.1
5	44713.7	29995.7	28346.5	29244.3	29684.7	258.7
6	47767.9	24557.8	23925.6	24543.0	24970.7	331.5
7	51706.1	27823.9	26664.6	27457.0	27833.1	246.4
8	49396.9	27432.0	25392.6	27049.5	27508.7	321.3
9	38514.0	26252.8	24783.2	25456.8	25961.0	369.3
10	36558.2	23957.7	22860.3	23739.2	23970.1	263.5
Avg.	45310.7	26343.8	25116.6	26197.0	26625.7	
St.d.	4618.1	1806.4	1671.3	1813.6	1772.9	

For the instances with 100 items and 20 alloys the difference between the solution provided by the proposed heuristics and CPLEX solver is huge and reached up to 80%. EBH is slightly better than CPLEX solving the model with the simplest

Lagrangian relaxation of the combined lot-sizing model. Again the rolling horizon approach performed the best, but only by 4% than EBH method.

5. Conclusions

In this paper we analyzed a practical problem of the production planning in a foundry, when both the capacity of the furnace in which alloy is produced and the capacity of core shop in which cores are prepared for further assembly in molds are limited.

Mixed integer programming model is developed based on the Capacitated Two-Level Lot Sizing and Scheduling Problem. Since CPLEX solver (12.7) was not able to provide good results with the industrial-size instances of the problem we provide different heuristic that were able to improve the initial solutions by up to 80%. Two of them are based on Lagrangian relaxation of the constraints related to the production planning of cores, one is based on the relax and fix approach for rolling horizon, and finally evolutionary based heuristic is proposed which does not require any commercial solver.

The tests, conducted on the instances with two different sizes of the problem (with 50 and 100 items, and 50 sub-periods) that are equivalent to the real industrial planning problems, show that the proposed solutions are comparable and differ at most by 10%. The relax and fix approach provides slightly better results than other heuristics, but if we are not willing to use any commercial MIP solver, proposed evolutionary based heuristic is a reasonable choice.

In the future work, we may extend our research to the Capacitated Two-Level Lot Sizing and Scheduling Problem with two melting furnaces and two automatic molding lines, which is often encountered in modern foundries. Moreover, consideration of the problem "make or buy" in a case of core shop limited capacity can be useful for economic practice and a challenge for theoretical work.

References

[1] de Araujo, S.A., Arenales, M.N. & Clark, A.R. (2008). Lot sizing and furnace scheduling in small foundries. *Computers & Operations Research*. 35(3), 916-932. DOI: 10.1016/j.cor.2006.05.010

[2] Beyer, H.G. & Schwefel, H.P. (2002). Evolution strategies – A comprehensive introduction. *Natural Computing*. 1(3), 3-52. DOI:10.1023/A:1015059928466

[3] Drexl, A. & Kimms, A. (1997). Lot sizing and scheduling - survey and extensions. *European Journal of Operational Research*. 99(2), 221-235. DOI: 10.1016/S0377-2217(97)00030-1

[4] Hans, E. & van de Velde, S. (2011). The lot sizing and scheduling of sand casting operations. *International Journal of Production Research*. 49(9), 2481-2499. DOI: 10.1080/00207543.2010.532913

[5] Karimi, B., Ghomi Fatemi, S.M.T. & Wilson, J.M. (2003). The capacitated lot sizing problem: a review of models and algorithms. *Omega*. 31(5), 365-378. DOI: 10.1016/S0305-0483(03)00059-8

[6] Stawowy, A. & Duda, J. (2012). Models and algorithms for production planning and scheduling in foundries - current state and development perspectives. *Archives of Foundry Engineering*. 12(2), 69-74. DOI: 10.2478/v10266-012-0039-4

[7] Stawowy, A. (2006). Evolutionary algorithm for manufacturing cell design. *Omega*. 34(1), 1-18. DOI: 10.1016/j.omega.2004.07.016