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

ARCHIVES of FOUNDRY ENGINEERING

DOI: 10.2478/afe-2014-0053

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

ISSN (2299-2944)
Volume 14
Issue 3/2014

17 – 20

Mathematical Programming for Lot Sizing and Production Scheduling in Foundries

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

Received 30.04.2014; accepted in revised form 15.05.2014

Abstract

The problem considered in the paper is motivated by production planning in a foundry equipped with the furnace and casting line, which provides a variety of castings in various grades of cast iron/steel for a large number of customers. The quantity of molten metal does not exceed the capacity of the furnace, the load is a particular type of metal from which the products are made. The goal is to create the order of the melted metal loads to prevent delays in delivery of goods to customers. This problem is generally considered as a lot-sizing and scheduling problem. The paper describes a mathematical programming model that formally defines the optimization problem and its relaxed version that is based on the conception of rolling-horizon planning.

Keywords: Application of information technology to the foundry industry, Production planning, Scheduling

1. Introduction

In this paper we studied a scheduling problem in a mid-size foundry that makes castings to clients order In this case, the production planning problem consists in determining the lot size of the items and the required alloys to be produced during each period of the finite planning horizon that is subdivided into smaller periods (work shifts). Decision maker must take into account two main criteria: timeliness of orders and maximization of production capacity. Assuming that a production bottleneck is the melting furnace, a mixed-integer programming (MIP) models are usually proposed to solve the outlined above lot-sizing and scheduling problem.

The aim of this paper is to explore whether Excel or commercial nonlinear solvers may be used successfully towards small and medium-sized foundries when planning and scheduling decisions are taken. Section 2 provides a MIP model for foundry scheduling problem. In Section 3, the details of proposed approaches are given. The computational experiments are described in Section 4, and the conclusions are drawn in Section 5.

2. Lot-sizing and scheduling model

2.1. MIP lot-sizing model

The MIP model presented in this section is an extension of Araujo et al. lot sizing and scheduling model for automated foundry [1]. We use the following notation:

Indices

i=1,…,I - produced items; k=1,…,K - produced alloys,
t=1,…,T - working days; n=1,…,N - sub-periods,

Parameters

d_{it} - demand for item i in day t; w_i - weight of item i,
a_i^k = 1, if item i is produced from alloy k, otherwise 0,
st_k - setup loss for alloy k; C - loading capacity of the furnace,
h_i^-, h_i^+ - cost for delaying (–) and storing (+) production of item I,
s - setup penalty (cost) when alloy is change in the furnace.

Variables
I_{it}^-, I_{it}^+ - number of items i delayed (–) and stored (+) at the end of day t,
z_n^k = 1, if there is a setup (resulting from a change) of alloy k in sub-period n, otherwise 0,
y_n^k = 1, if alloy k is produced in n in sub-period, otherwise 0,
x_{in} - number of items i produced in sub-period n.

Production planning problem in a foundry is defined as follows:

$$\text{Minimize} \quad \sum_{i=1}^{I}\sum_{t=1}^{T}(h_i^- I_{it}^- + h_i^+ I_{it}^+) + \sum_{k=1}^{K}\sum_{n=1}^{N}(s \cdot z_n^k) \tag{1}$$

subject to:

$$I_{i,t-1}^+ - I_{i,t-1}^- + \sum_{n=1}^{N} x_{in} - I_{it}^+ + I_{it}^- = d_{it}, \quad i = 1,...,I, t = 1,...,T \tag{2}$$

$$\sum_{i=1}^{I} w_i x_{in} a_i^k + st_k z_n^k \le C y_n^k, \quad k = 1,...,K, n = 1,...,N \tag{3}$$

$$z_n^k \ge y_n^k - y_{n-1}^k, \quad k = 1,...,K, n = 1,...,N \tag{4}$$

$$\sum_{k=1}^{K} y_n^k = 1, \quad n = 1,...,N \tag{5}$$

$$I_{it}^-, I_{it}^+, x_{it} \ge 0, \quad I_{it}^-, I_{it}^+, x_{it} \in \Im, \quad I_{i0}^-, I_{i0}^+ = 0, \ i = 1,...,I \tag{6}$$

The goal (1) is to find a plan that minimizes the sum of the costs of delayed production, storage costs of finished goods and the setup costs for alloy changing in the furnace.

Equation (2) balances inventories, delays and the volume of production of each item in each day. Constraint (3) ensures that the furnace capacity is not exceeded in a single load. Constraint (4) sets variable z_n^k to 1, if there is a change in an alloy in the subsequent periods, while constraint (5) ensures that only one alloy is produced in each sub-period.

The model itself can be seen as an extension of generalised lot-sizing and scheduling problem (GLSP) that is well described in literature and for which standard MIP methods usually achieve acceptable results [2,3]. However, since the lot-sizing model for a foundry takes into account also the order of alloys – setup penalty is calculated as a part of the objective function (1) and alloy changing loss is included in constraint (3) – it is much harder to solve than the classic lot-sizing model.

2.2. Relaxed lot-sizing model

The model presented in the previous section, even for the smallest problem considered later in the experiments and containing 10 items, 2 different alloys and the planning horizon of 50 sub-period in total, results in 830 optimization variables (210 binary and 630 integer). This is far too many Excel Solver can handle (the limit for standard built-in MS Excel Solver is 200 variables). Thus we decided to apply a method similar to the fix and relax method proposed by Araujo et al. in [1]. The main idea behind this method is to compute the exact plan only for a single day, while for remaining days only rough plan is determined. This is called rolling-horizon planning [4].

Araujo et al. [1] relaxed all 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 is of a float type instead of integer, while y_t^k is now integer value reflecting in how many sub-periods a given alloy will be produced.

Thus constraint (2) is valid only for the fixed day (t_f) and for other days it looks as follows:

$$\sum_{i=1}^{I}\sum_{n=1}^{N} w_i x_{in} a_i^k \le C y_t^k, \quad k = 1,...,K, t = 1,...,T, t \ne t_f \tag{7}$$

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

$$\sum_{k=1}^{K} y_t^k = N, \quad t = 1,...,T, t \ne t_f \tag{8}$$

Instead of one model computed once, the model for rolling-horizon is computed T times. Each time values of fixed variables computed for one day are included as constants in the following days.

In order to make the model possible for solving by MS Excel Solver we decided to farther reduce the size of the model. For non-fixed days we replaced variables x_{in} with x_{it}, what means that the production equation (2) for the relaxed days is balanced as:

$$I_{i,t-1}^+ - I_{i,t-1}^- + x_{it} - I_{it}^+ + I_{it}^- = d_{it}, \quad i = 1,...,I, t = 1,...,T, t \ne t_f \tag{9}$$

This allows for the reduction of variables to only 140 integer variables (100 for the fixed day and 40 for the remaining days). Variable y_n represents the alloy grade produced in a given sub-period (only for the fixed day). Constraint (8) is then not used, and only total capacity of furnace is checked for the relaxed days:

$$\sum_{i=1}^{I} w_i x_{it} + st_k \le C, \quad k = 1,...,K, t = 1,...,T \tag{10}$$

This allows for farther reduction of binary variables from 130 to only 10, giving the total number of 150 variables.

3. Solution methods

Experiments performed by the authors [5] for large instances of lot-sizing and scheduling problem indicated that such problems are hard to solve and it is worth to apply computational intelligence methods like genetic algorithms. In this paper we want to examine whether Excel Solver can be used for small instances of the problem and what is the impact of problem relaxation on the final result.

3.1. Excel Solver

Excel Solver is provided by Frontline Systems. The firm also sales an extended version of the solver, which enables to solve larger problems. For non-linear problem, like ours, it can handle 500 variables and 250 constraints (compare [6]), while the version built-in MS Excel application is limited to 200 variables and 100 constraints. We decided to use Frontline Solver Pro, as we found that it performs better than standard Excel Solver even for the same amount of variables and constraints. This solver is continuously updated, while the built-in MS Excel version changes only with a new version of application (usually once every three years).

Regarding the solver limitation for non-linear problems we were able to compute only the smallest example with 10 items and 2 different alloys. The problem with 50 items and 10 alloys would require using 710 variables, what exceeds the limit.

The main fragment of a spreadsheet that calculates production plan in a foundry is shown in Fig. 1. First table contains input data, while the second one contains a detailed plan for one day (with 10 sub-periods) and rough plan for the remaining days (2-5).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	**Lot-sizing and scheduling**														
2	**Input data**			coef.	basic	result	min load								
3	Furnace capacity [kg]			1,1	629	**691,9**	518,93								
4	Setup penalty			5			0,75								
5	Setup lost for alloy 1 and			1	9										
6				2	5										
7		Inventory					Demand per day								
8	Castings	Quantity	Alloy	Weight [kg]	Holding cost	Delaying cost/day	**1**	**2**	**3**	**4**	**5**				
9	C1	4	1	8,0	0,7	4,8	57	28	10	35	31				
10	C2	-2	1	16,0	0,8	7,8	24	50	46	25	26				
11	C3	4	1	5,0	0,6	7,0	21	60	11	38	52				
12	C4	-10	1	17,0	0,8	5,0	14	10	53	42	10				
13	C5	7	1	12,0	0,7	7,9	34	58	32	33	21				
14	C6	9	2	2,0	0,5	3,8	39	20	16	29	44				
15	C7	-2	2	22,0	0,9	6,5	22	45	26	42	48				
16	C8	1	2	20,0	0,9	8,7	31	41	12	40	50				
17	C9	-10	2	30,0	1,1	4,7	60	47	58	45	48				
18	C10	10	2	28,0	1,1	6,6	44	41	45	49	59				
19															
20	**Produced alloy and castings/sub-period**														
21		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**2**	**3**	**4**	**5**
22		1	1	1	1	1	1	1	1	1	1				
23	1	1	0	1	0	1	0	0	0	0	0				
24	2	0	1	0	1	0	1	1	1	1	1				
25	C1	2	3	52	4	0	1	1	2	0	2	27	10	39	28
26	C2	3	2	1	3	22	1	1	1	1	1	50	46	25	26
27	C3	2	1	2	2	14	1	0	2	0	1	59	11	38	52
28	C4	12	1	3	1	9	0	1	0	1	0	10	53	47	5
29	C5	20	2	2	2	5	1	0	1	0	1	71	19	36	21
30	C6	6	5	5	1	3	1	5	1	2	15	20	16	29	44
31	C7	3	4	3	1	2	7	8	1	0	4	44	26	42	48
32	C8	3	0	4	3	3	0	9	3	13	2	41	12	40	50
33	C9	9	0	18	17	38	14	1	11	12	16	46	58	45	48
34	C10	4	18	5	1	4	1	6	6	2	0	41	45	49	59

Fig. 1. Fragment of a spreadsheet that calculates production plan

3.2. CPLEX solver

We wrote the model in optimization programming language (OPL) that was later run in IBM CPLEX Optimization Studio 12.6. The basic model in OPL language is shown in Fig. 2.

```
minimize
 sum(i in ir, t in tr) (hm[i]*im[i][t]+hp[i]*ip[i][t])
 + sum(k in kr, t in tr, n in nr) sp*z[k][t][n];
subject to
{
 forall(i in ir) im[i][0]==0;
 forall(i in ir) ip[i][0]==0;
 forall(k in kr,t in tr, n in nr) y[k][t][0]==0;
 forall(i in ir,t in tr) ip[i][t-1]-im[i][t-1]+ sum(n in nr)
  x[i][t][n]-ip[i][t]+im[i][t] == d[i][t];
 forall(k in kr, t in tr, n in nr) sum(i in ir)
  w[i]*x[i][t][n]*a[i][k]+st[k]*z[k][t][n]<=C*y[k][t][n];
 forall(k in kr, t in tr, n in nr) sum(i in ir)
 forall(k in kr, t in tr,n in nr)
   z[k][t][n]>=y[k][t][n]-y[k][t][n-1];
 forall(t in tr, n in nr) sum(k in kr) y[k][t][n]==1; };
```

Fig. 2. Basic OPL model for foundry production planning

Arrays *hm*, *hp* represent costs h_i^- and h_i^+, while arrays *im* and *ip* represents variables I_{it}^- and I_{it}^+, respectively. Variables *x*, *y* and *z* are indexed by both day (*t*) and its sub-periods (*n*). Sub-period number that is used in the model 2.1 can be calculated as: $t*N+n$.

3.3. CPLEX solver with relaxation

For the largest problem considered in the experiments CPLEX was not able to provide optimal solution for the model. The solution - even after 10 minutes - was on average 50-70% distant from the theoretical lower bound, so we decided to experiment with the fixed and relaxed approach described in section 2.2. The objective function remains the same (however this time it is calculated in *T* steps). The constraints in the rolling horizon version of the model written in OPL language are shown in Fig. 3.

```
subject to
{
 forall(i in ir) im[i][0]==0;
 forall(i in ir) ip[i][0]==0;
 forall(k in kr, t in tr) y[k][t][0]==0;
 forall(i in ir, t in fixed_periods, n in nr) xr[i][t][n]==0;
 forall(i in ir, t in relaxed_periods, n in nr) x[i][t][n]==0;
 forall(k in kr, t in relaxed_periods, n in nr) y[k][t][n]==0;
 forall(k in kr, t in fixed_periods) yr[k][t]==0;
 forall(i in ir,t in tr) ip[i][t-1]-im[i][t-1]+ sum(n in nr)
  (x[i][t][n] + xr[i][t][n])-ip[i][t]+im[i][t] == d[i][t];
 forall(k in kr, t in fixed_periods, n in nr) sum(i in ir)
  w[i]*x[i][t][n]*a[i][k]+st[k]*z[k][t][n]<=*C*y[k][t][n];
 forall(k in kr,t in relaxed_periods) sum(i in ir, n in nr)
  w[i]*xr[i][t][n]*alloy[i][k]<=C*yr[k][t];
 forall(k in kr, t in fixed_periods, n in nr)
  z[k][t][n]>=y[k][t][n]-y[k][t][n-1];
 forall(t in fixed_periods, n in nr) sum(k in kr)
  y[k][t][n] == 1;
 forall(t in relaxed_periods) sum(k in kr) yr[k][t] == N; };
```

Fig. 3. Constraints in rolling-horizon model for foundry production planning

Arrays *xr* and *yr* represent relaxed variables x_{in} and y_t^k, respectively. *Fixed_periods* are changed from 1,…,1, to 1,…,5 and *relaxed_periods* are changed from 2,…,5 to empty range in the last iteration.

In order to automate the computing process we wrote C# application that run 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 3 minutes, however usually the partial solution was provided within 1 minute or almost immediately, so the final solution after 5 days was usually achieved within 6-7 minutes.

4. Computational experiments

4.1. Test problems

Computational experiments were conducted on the basis of the test problems proposed in [1]. The characteristic of these problems is covered in Table 1. The values for demand and weight were determined using uniform distribution within a given range.

Table 1.
Test problems characteristics

Parameter	Value
number of items (I), number of alloys (K)	(10,2); (50,10); (100,20)
number of days (T)	5
number of subperiods (N)	10
demand (d_{it})	[10, 60]
weight of item (w_i)	[1, 30]
setup loss for alloy (st_k)	[5, 10]
setup penalty (s)	5

We extended the test problems by introducing costs for delayed items and costs for storing the items produced that do not match the demand. Those parameters were randomly generated as follows:

$$h_i^- = 6 * random_{01} + 3 \tag{11}$$

$$h_i^+ = w_i * 0.02 + 0.05 \tag{12}$$

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

$$C = \frac{\sum_{i=1}^{N}\sum_{t=1}^{T} d_{it} w_i + \sum_{k=1}^{K} st_k}{50} \tag{13}$$

4.2. Results of the experiments

Computational experiments were conducted for three sizes of planning problems: with 10 items made from 2 different alloys, 50 items made from 10 alloys, and with 100 items and 20 alloys. For each problem size ten instances were computed. Excel Solver was able to handle only the smallest problem, regarding the limit of optimization variables. The computational time was similar for all the examined methods and ranged from 6 to 8 minutes.

The average results achieved for all 10 instances along with the standard deviations are presented in Table 2.

Table 2.
Results of the experiments for examined methods

Problem		Excel	CPLEX	CPLEX-RH
(10,2)	**average**	**151.85**	**54.54**	**68.03**
	std.dev.	66.82	13.51	23.87
(50,10)	**average**	–	**6,794.15**	**6,254.24**
	std.dev.		1,292.54	1,021.46
(100,20)	**average**	–	**34,695.10**	**29,038.12**
	std.dev.		2,339.14	1,961.16

The experiments for 10 items and 2 alloys clearly show that the solution provided by Excel cannot compete with the solution achieved by the advanced MIP solver that is commercially available from IBM. Nevertheless, our goal was to demonstrate that when model is appropriately relaxed Excel can be used as a basic and cheap tool for optimization of production planning in a foundry. However, when more variables need to be covered in the model or more accurate solution is desired advanced MIP solvers like CPLEX should be recommended.

For the larger instances of the problem fixed and relax method based on the rolling-horizon planning gave on average better results than when the planning model was optimized for all the periods at once. This is due to the fact that a single relaxed problem can be solved optimally or close to optimal, while for the non-relaxed problem the solver usually delivers solutions that can be even 70% distant from the theoretical lower bound.

5. Conclusions

In this paper the mathematical programming is applied for foundry production planning. The model is based on a well-known lot-sizing problem that was extended to handle the constraints regarding changes in alloy grade. Such a model is difficult to solve as it includes large amount of decision variables (few thousand for the problem of a medium size). The number of variables can be reduced by applying the concept of rolling-horizon planning. In such approach variables are computed precisely only for one period (e.g. a day), while for remaining periods (days) variables are roughly computed in order to satisfy the constraints. Although such relaxed problem usually does not allow to find optimal solution it can provide good approximation of optimal solution and within shorter computational time.

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] 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.

[3] Fleischmann, B., Meyr, H. (1997). The general lot sizing and scheduling problem. *Operational Research Spektrum*. 19(1), 11-21. DOI: 10.1007/BF01539800.

[4] Clark, A.R. (2005). Rolling horizon heuristics for production and setup planning with backlogs and error-prone demand forecasts. *Production Planning & Control*. 16, 81-97. DOI: 10.1080/09537280412331286565.

[5] Stawowy, A. & Duda, J. (2013). Production scheduling for the furnace - casting line system. *Archives of Foundry Engineering*. 13(3), 84-87. DOI: 10.2478/afe-2013-0065.

[6] http://www.solver.com