ARCHIVES of FOUNDRY ENGINEERING

ISSN (1897-3310)
Volume 15
Special Issue
2/2015

3 – 6

1/2

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

Production Scheduling for the One Furnace – Two Casting Lines System

R. Basiura, J. Duda *, A. Stawowy
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 06.05.2015; accepted in revised form 01.06.2015

Abstract

The paper presents a scheduling production problem in foundry equipped with one furnace and two casting lines, which provides a number of different types of castings for a large number of clients. The amount of molten metal may not be greater than the capacity of the furnace and its load is a type of metal, from which the products are manufactured on automated casting lines. The purpose of planning is to create the processing order of metal, to prevent delays in the delivery of the ordered products to the customers. This problem is mixt of lot-sizing problem and scheduling problem on two machines (the lines) running in parallel. The article gives a mathematical model, which formally defines the optimization problem, and his relaxed version which is based on the concept of rolling-horizon planning. The proposed approaches were tested on the sample data.

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 employing make-to-order (MTO) strategy to provide several types of metal alloys in lots for a large number of customers. In this case, the production planning problem consists in determining the lot size of the ordered items and the required alloys to be produced during each period of the finite planning horizon that is subdivided into smaller periods (e.g. furnace loads). Decision maker must take into account two main criteria: satisfaction of orders' due dates and maximization of production capacity. A mixed-integer programming (MIP) models are usually proposed to solve this lot-sizing and scheduling problem.

The aim of this paper is to present the effective methods for production planning and scheduling in the one furnace-two casting lines system. 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. MIP lot-sizing and scheduling model

The MIP model presented in this section is an extension of Araujo et al. lot sizing and scheduling model for automated foundry [1]. The model takes into account the assumption that the casting lines can run partially in parallel, i.e. a part of the products can be manufactured on both lines, while other casts may be prepared either on the first line or on the other one. Also two additional technological constraints are introduced: the furnace load must not be less than a specified quantity, as well as the production of individual castings on both lines must be greater than the lower limit. We use the following notation:

Indices
$i=1,\dots,I$ - produced items; $k=1,\dots,K$ - produced alloys
$t=1,\dots,T$ - working days; $n=1,\dots,N*T$ - sub-periods, where N number of sub-periods per day
$l=1, 2$ casting line number.

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

m_{il} - the minimum lot size of item i on line l. When production of item i is not allowed on line l, then m_{il}=0.

C - loading capacity of the furnace

h_i^-, h_i^+ - cost for delaying (–) and storing (+) production of item I,

s - setup penalty when alloy is changed 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 sub-period n, otherwise 0

x^l_{in} - number of items i produced in sub-period n on line l.

Production planning problem in a foundry is defined as follows:

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

subject to:

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

$$\sum_{i=1}^{I} w_i (x_{in}^1 + x_{in}^2) a_i^k \le C y_n^k \quad k=1,...,K, n=NT \quad (3)$$

$$z_n^k \ge y_n^k - y_{n-1}^k, \quad k=1,...,K,\ n=1,...,NT \quad (4)$$

$$\sum_{k=1}^{K} y_n^k = 1, \quad n=1,...,NT \quad (5)$$

$$0 \le x_{in}^l \cdot (x_{in}^l - m_{il}), \quad i=1,...,I, n=1,...,NT, l=1,2 \quad (6)$$

$$C\cdot m_{il} \ge x_{in}^l, \quad i=1,...,I, n=1,...,NT, l=1,2 \quad (7)$$

$$\begin{aligned} &I_{it}^-, I_{it}^+, x_{it} \ge 0, \quad I_{it}^-, I_{it}^+, x_{it} \in \Im, \quad I_{i0}^-, I_{i0}^+ = 0,\ i=1,...,I \\ &0 \le z_n^k \le 1, n=1,...,NT, k=1,....,K \end{aligned} \quad (8)$$

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 penalty 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. Constraint (6) provides that lot size of item i is large enough. Constraint (7) ensures that items not allowed on line l are not produced on that line.

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 [4][5]. 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) – it is much harder to solve than the classic lot-sizing.

We decided also to employ a fix and relax method proposed by Araujo et al. in [1]. The main idea of this method is to compute the exact plan only for a single day, while only rough plan is determined for remaining days of planning period. This is called rolling-horizon planning [2].

In the fixed and relax method all variables x_{in} and y_n^k for the sub-periods that does not belong to the fixed day are relaxed. Variable x_{in} representing the number of items i produced in sub-period n is for relaxed periods now contains float type values instead of an integer ones, and variable y_t^k for the relaxed periods now contains integer values representing the number of sub-periods in which a given alloy is produced.

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

$$\sum_{i=1}^{I} w_i (x_{it}^1 + x_{it}^2) a_i^k \le C y_t^k, \quad k=1,...,K, t=1,...,T, t \ne t_f \quad (9)$$

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

$$\sum_{k=1}^{K} y_t^k = N, \quad t=1,...,T, t \ne t_f \quad (10)$$

The model for rolling-horizon is computed T times. Each time values of variables x_{in} and y_n^k computed for the fixed periods are included as constrains in the following days.

3. Solution methods

In this paper we used the same approach that has been described in [3]. Our goal was to find whether a significantly more complex problem can still be solved by Excel Solver and how efficient it would be, comparing to the solution achieved by CPLEX Solver. Additional goal was to assess the benefits from using rolling horizon method, as the relaxation of decision variables should bring even more improvement than for the model with only one casting line.

MS Excel with its adds-on Solver is very popular tool for simple decision problems. The main limitation of Excel Solver is its ability to handle only 200 variables and 100 constraints. The model presented in Section 2, even for the smallest problem considered (with 10 items, 2 different alloys and the planning horizon of 6 sub-periods) in total, results in 630 optimization variables (30 binary and 600 integer). Moreover, models using nonsmooth Excel functions (e.g., IF, MAX, MIN, ABS) are very hard to solve. That's while we tried to use open source OpenSolver [8] and commercial Frontline Solver Pro [6], as they can handle problems of virtually unlimited size.

It turned out, however, that even for the smallest of examined problems is not possible to achieve a solution using OpenSolver engines due to the nonlinearity and discontinuity of an Excel model. In case of Frontline Solver Pro, the Standard GRG (Generalized Reduced Gradient) engine [7] was able to partially perform its task, but only for the smallest problems. Partially - as GRG Solver could not find a feasible solutions: the decision variables were not integers and all constraints have not been met.

We then use the following procedure:

1. Generate initial solution based on the knowledge of the problem.
2. Start the GRG engine.
3. When decimal solution was obtained, round the values of decision variables to nearest integers.

4. Start GRG once again.
5. Perform rounding as in (3) to obtain a final solution.

The solutions obtained in this way never met the constraints with the number of available goods greater than or equal to the demand. Such unsuccessful experiment proves what a difficult problem we have to deal with, and, on the other hand, testifies the weakness of the proposed, sometimes expensive add-in solvers.

The only solution that could be apply is the concept of a rolling horizon planning, described earlier. The authors have already studied such approach for a small foundry with a single casting line [3].

In the next stage of the experiments we wrote the model in optimization programming language (OPL). In order to write the model in an form appropriate for CPLEX we need to introduce additional binary variable (*xb*) to properly handle constraint (6) that ensures that either no item *i* is produced in a given subperiod or the production lot for this time exceeds a minimal lot-size. We also had to change constraint (7) and to introduce additional constraint to make sure *xb* variable is properly determined ($xb_{in}^{l}=0$ if item *i* not scheduled for a line *l* in the subperiod *n*; 1 – otherwise). New constraints (6) and (7) look now as follows:

$$x_{in}^{l} \geq m_{il} xb_{in}^{l}, \quad i=1,...,I, n=1,...,NT, l=1,2 \qquad (11)$$

$$xb_{in}^{l} \cdot M \geq x_{in}^{l}, \quad i=1,...,I, n=1,...,NT, l=1,2 \qquad (12)$$

We run the model in the latest version of IBM CPLEX Optimization Studio (12.6.1). We let the solver to optimize the problem within a given time limit, but half of it was devoted to polishing the solution to improve the best known solution at the end of branch and cut procedure.

4. Computational experiments

4.1. Test problems

Computational experiments were conducted on the basis of the test problems prepared by the authors. Three sizes of planning problems were considered: with 10 items made from 2 different alloys, 20 items made from 5 alloys, and finally with 50 items and 10 alloys. The characteristic of these problems is covered 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)	(10,2); (20,5); (50,10)
number of days (T)	5
number of subperiods (N)	6
demand (d_{it})	[10, 60]
weight of item (w_i)	[2, 50]
setup penalty (s)	10
delaying cost (h_i^-)	[3.00, 9.00]
holding cost (h_i^+)	w_i * 0.02 + 0.05

Three 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:

$$C = \frac{\sum_{i=1}^{N} \sum_{t=1}^{T} d_{it} w_i + \sum_{k=1}^{K} st_k}{30} \qquad (13)$$

4.2. Results of the experiments

For each problem size three instances were computed. GRG solver for Excel was able to handle only the smallest problem, regarding the limit of optimization variables. The computational time for Excel Solver was about 10 minutes, while for CPLEX generating a schedule for 5 days in a single run we set a time limit to 180 sec., and for rolling-horizon we set a time limit to 60 sec. for each single run. However also in the latter case overall computational time usually did not exceed 180 sec.

The results achieved for all test instances in computational experiments are summarized in Table 2. For Excel solver the average result along with standard deviation from five runs is shown. Column denoted as RH contains the results achieved by rolling-horizon method run in CPLEX solver, while the 'LB' column shows a theoretical lower bound for the objective function that was also provided by CPLEX solver.

Table 2.
Results of the experiments for different methods

Problem	Test	Excel	CPLEX	RH	LB
(10,2)	1	1 066.90 (311.08)	149.14	149.14	148.93
	2	1 357.24 (57.96)	105.94	108.99	105.92
	3	1 146.24 (23.5)	120.55	121.58	119.41
(20,5)	1		1 878.64	1 860.83	1 145.61
	2		4 797.44	4 758.88	4 183.37
	3		641.51	558.31	437.47
(50,10)	1		7 814.18	7 620.40	3 149.02
	2		5 807.53	5 612.05	2 891.44
	3		8 366.86	8 095.69	1 523.28

The experiments for 10 items and 2 alloys clearly show that even sophisticated Excel solver cannot efficiently deal with the two casting lines problem stated in the paper. Thus, due to their limitation, such solvers cannot be used as a basic and cheap tool for optimization of production planning in a foundry. In such case a dedicated heuristic, e.g. based on one of the computational intelligence methods, or mixed integer programming solvers like CPLEX have be used.

CPLEX solver for both single run and rolling horizon approaches achieved good results, however, for the problems with

50 items, the solution was at best 50% distant from a theoretical lower bound. For the last test problem this difference exceeded as much as 80%. Apart from the smallest test problem instances (with 10 items), the fixed and relax method based on the rolling-horizon planning gave better results than when the planning model was optimized for all days at once. However the difference was only maximum 3%, which was far above the gain we expected before the experiments. This indicates, that the research should be continued aiming at developing new ways to simplify the problem.

5. Conclusions

In this paper, the mathematical programming model presented earlier [3] for a small foundry has been extended to the more complex problem with two casting lines that can be found usually in a medium-sized foundry. As before, the model is based on a well-known lot-sizing problem extended to handle the constraints regarding changes in alloy grade and the presence of two manufacturing lines. We have shown that such model is very difficult to solve as it includes huge amount of optimization variables (few thousand for the problem of a medium size). The number of variables can be reduced by applying the conception of rolling-horizon planning. In such approach variables are computed precisely only for one period (a day), while for next days, variables are roughly computed in order to satisfy the constrains. However, as it has been shown for the smallest instances of the problem (with 10 items and 2 alloys) such relaxed problem usually does not allow to achieve the optimal solution. Nevertheless, it can provide good approximation of optimal solution in a shorter computational time.

The computational experiments presented in the paper prove that both open-source and even quite expensive commercial solvers for Excel cannot be used as a reliable tool for optimization of production planning in a medium-sized foundry. Their limitations as to the size of the problem and the impossibility of dealing with discontinuous and nonlinear functions make them impractical in real-life scheduling problem solving.

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] 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.

[3] Duda, J., Stawowy, A. & Basiura, R. (2014). Mathematical programming for lot sizing and production scheduling in foundries. *Archives of Foundry Engineering*. 14(3), 17-20. DOI: 10.2478/afe-2014-0053.

[4] 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.

[5] Fleischmann, B. & Meyr, H. (1997). The general lotsizing and scheduling problem. *Operational Research Spectrum*. 19(1), 11-21. DOI: 10.1007/BF01539800.

[6] FrontlineSolvers, http://www.solver.com/premium-solver-platform.

[7] Lasdon, L.S., Fox, R.L., & Ratner, M.W. (1974). Nonlinear optimization using the generalized reduced gradient method. *RAIRO - Operations Research - Recherche Opérationnelle*. 8(3), 73-103.

[8] OpenSolver for Excel, http://opensolver.org.