A R C H I V E S O F M E T A L L U R G Y A N D M A T E R I A L S
Volume 58 2013 Issue 3
DOI: 10.2478/amm-2013-0088

J. DUDA*, A. STAWOWY*

OPTIMIZATION METHODS FOR LOT-SIZING PROBLEM IN AN AUTOMATED FOUNDRY

ALGORYTMY PLANOWANIA PARTII PRODUKCYJNYCH W ZAUTOMATYZOWANEJ ODLEWNI

In the paper we studied a production planning problem in a mid-size foundry that provides tailor-made cast products in small lots for a large number of clients. Assuming that a production bottleneck is the furnace, a mixed-integer programming (MIP) model is proposed to determine 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. As using an advanced commercial MIP solvers may be impractical for more complex and large problem instances, we proposed and compared a few computational intelligence heuristics i.e. tabu search, genetic algorithm and differential evolution. The examination showed that heuristic approaches can provide a good compromise between speed and quality of solutions and can be used in real-world production planning.
Keywords: Application of information technology to the foundry industry, Production planning, Lot-sizing problem

W pracy przedstawiono problem planowania produkcji w odlewni średniej wielkości, która dostarcza odlewy na zamówienie dla dużej liczby klientów. W takim problemie konieczne jest określenie wielkości partii produkcyjnej oraz ilości i gatunku metalu w każdym okresie skończonego horyzontu planowania, który jest podzielony na mniejsze podokresy. Przy założeniu, że wąskim gardłem jest piec do topienia metalu, zaproponowano programowanie całkowitoliczbowe mieszane (Mixed-Integer Programming – MIP) jako model planowania i harmonogramowania produkcji w odlewni. Jako że użycie zaawansowanych komercyjnych solverów może być niepraktyczne dla złożonych problemów, zaproponowano i porównano trzy heurystyki inteligencji obliczeniowej tj. tabu search, algorytm genetyczny i ewolucja różnicowa. Eksperymenty obliczeniowe wykazały, że algorytmy heurystyczne zapewniają zadowalającą szybkość i jakość rozwiązań.

1. Introduction

In this paper we studied a production planning problem in a mid-size foundry that provides tailor-made cast products employing several types of metal alloys in small lots for a large number of clients. In such production planning problem, it is necessary to determine 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 maximizing production capacity bottlenecks. The stated problem is described by lot scheduling models, from which the most explored is Mixed Integer Programming model [7]. As exact commercial solvers are impractical for large problem instances, we proposed to apply some heuristic search methods to solve the MIP model.

The aim of this paper is to explore whether computational intelligence techniques may be used successfully towards small and medium-sized foundries when planning and scheduling decisions are taken. Our paper is organized as follows. Section 2 provides a literature review on foundry lot-sizing and scheduling problem. In Section 3, the details of proposed model and heuristics are given. The computational experiments are described in Section 4, and the conclusions are drawn in Section 5.

2. Review of previous research

There are only a few studies reported on the lot-sizing problem for production planning and scheduling in foundries.

Santos-Meza at al. [6] have studied a lot-sizing problem in an automated foundry when the production bottleneck is the furnace. The problem consists of two decisions: what alloys should be produced in the furnace in each period, and the quantity of items to be produced in each moulding machine. The objective function is to minimize the total production costs. The authors have proposed problem-specific heuristic to solve large practical integer programming problem. In this case some production constraints are relaxed without loss of optimality.

Araujo et al. [1] have dealt with the same problem prevalent in small market-driven foundries. A fast relax-and-fix (RF) approach is formulated to solve real-life instances in reasonable computing time: at the R step all integer variables are relaxed and the relaxed problem is solved using local search heuristics developed by authors, while at the F step partially

* AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY, FACULTY OF MANAGEMENT, GRAMATYKA 10, 30-067 KRAKOW, POLAND

Unauthenticated | 89.73.89.243
Download Date | 12/6/13 3:00 PM

fixed problem is solved to optimality with the CPLEX MIP solver.

Park and Yang [5] have introduced a Linear Programming (LP) optimization model for casting scheduling in job-shop type foundries. The primary variable in scheduling model is the amount of castings produced in each shift; the authors have argued that the objective function to maximize alloy utilization percentage and entire constraints reflecting real casting conditions can be represented in linear forms. The LP model generates an optimal casting sequence resulting in the maximum use of molten alloy.

Gauri [3] demonstrated that the product-mix planning problem for batches of melt can be modeled mathematically. Weighted integer goal programming formulations have been developed to determine the optimal product-mix for the immediate next heat in a small scale iron foundry, which can be useful to general iron foundries.

More recently, Camargo at al. [2] have considered the production planning problem in small foundries. The authors have proposed the heuristic that solves the problem in a hierarchical way. A genetic algorithm is used to explore a larger set of alloy sequences and a knapsack problem algorithm determines the lot size of the items for each furnace loading. The computational experiments show that the proposed approach is better than the methods described in the literature.

Teixeira at al. [8] have proposed a binary integer model for production scheduling problems in market-driven foundries. The objective is to minimize the cost of manufactured products based on balancing and synchronizing the moulding, pouring and finishing steps, aimed at eliminating high stock levels, rationing the use of production resources and foundry furnaces. Synchronization among the three production phases directly reduces production cycle time and indirectly improves quality of products.

3. Modelling approach

The mathematical model presented in this section results from an extension of Araujo et al. lot sizing and scheduling model for automated foundry [1].

3.1. Model description

We use the following notation to model production planning in a foundry:

Indices

$i = 1, ..., I-$ produced items; $k = 1, ..., K$ – produced alloys

$t = 1, ..., T-$ working days; $n = 1, ..., N$ – sub-periods (i.e. furnace loadings)

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 of capacity (kg) resulting from a setup for alloy K

s_k – setup penalty for alloy k

C – loading capacity of the furnace

h_{it}^-, h_{it}^+- penalty for delaying (–) and storing (+) production of item i in day t

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 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 problem as a MIP formulation can be defined as follows:

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

subject to:

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

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

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

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

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

The goal (1) is to find a plan that minimizes the cost of delayed production, storage of finished goods and the cost of setup times if the alloy is changed during furnace load.

Equation (2) allows for calculation of delayed and stored production at the end of a given period. Constraint (3) ensures that the furnace capacity is not exceeded during a single load. Constraint (4) sets variable z_n^k to 1, if there is a change in alloys in the subsequent periods, while constraint (5) ensures that only one alloy is produced in a given sub-period.

3.2. Solution methods

The planning problem for a foundry was intentionally written as MIP formula in order to compare two main approaches of finding solution for real-world production problems. In the first approach a branch and cut (B&C) method and its derivatives are usually used. If the problem is well constrained and the MIP formulation is adequate, this method is able to find optimal solution almost instantly or, for larger problems, in a reasonable time. However, if a problem is large and complex (e.g. belongs to NP-hard class) or it cannot be easily written as MIP formulation, either constraint programming (CP) or various heuristics can be applied as an alternative approach to the B&C method. Constraint programming uses such techniques like constraint propagation with backtracking to continuously refine the solution space. Among the heuristic methods, the ones based on metaheuristics like simulated annealing, genetic algorithms, tabu search are frequently used, as usually they allow to achieve a good approximation of the optimal solution in acceptable time. In the experiments that will be described in the following section, three metaheuristics are used: genetic algorithm (GA), tabu search (TS) and one of the youngest metaheuristics that was proved to be very efficient for many problems – differential evolution (DE). An extensive description of them can be found in [4]. We will focus on the modifications that have been introduced to the

Unauthenticated | 89.73.89.243
Download Date | 12/6/13 3:00 PM

standard algorithms: all the modifications has been applied after a large number of experiments, and the ones given the best results have been finally chosen.

Proposed GA uses a special representation of solution that is based on three vectors (chromosomes): a vector $\bar{x}$ representing the number of items that are produced in a given period, a vector $\bar{o}$ representing the order numbers of the produced items, and a vector $\bar{a}$ representing alloy number that is produced during this period. Such representation allowed for the development of three different mutation operators (perturbation schemes), that were applied to the solutions in a new population: the first mutation (applied with the probability p_{m1} = 0.02) simply adds or subtracts 1.0 for the randomly chosen element in the $\bar{x}$ vector, the second mutation (p_{m2} = 0.002) randomly exchanges two order numbers in the solution (provided that they are made from the same alloy), and the third mutation (p_{m3} = 0.002) randomly changes an alloy number and then adjusts the order numbers to this new alloy. In order to reduce solution space, the length of vectors is set to *N*MAX_CHANGES*, where *MAX_CHANGES* represents the maximum number of orders that are allowed to be produced in a single sub-period. The GA algorithm uses standard one point crossover (adjusted to the representation, applied with p_c = 50% probability) and binary tournament as a selection scheme. Population size is set to 50 individuals.

Our tabu search (TS) algorithm utilises the same representation as the GA (standard representation with all x_{in} variables provided worse results) and uses three kind of neighbourhood search that are analogous to the mutations proposed for the GA. Each type of perturbation has its own tabu list that blocks the moves at the same position in the solution for a given number of iterations ($tabu_{size}$ = *T*N*). No aspiration function is used.

Finally, proposed differential evolution (DE) algorithm uses the representation analogous to the MIP solver (with all x_{in} variables) and applies *RandToBest/2/Bin* strategy, as it occurred to be the most efficient in the experiments. This means that new solution is created on the basis of the best solution found so far and three other, randomly chosen solutions. As the binomial crossover may produce infeasible solutions (i.e. more than one alloy is produced in the given period n), a special repair algorithm is applied. If orders with the different alloys are planned for production in a period n, the one with the highest number of items is chosen, and the orders requiring different alloy are removed from the plan. Population size is set to 30 individuals, scale parameter for mutation is set to 0.7 and the crossover rate (p_{CR}) to 0.1. In order to assess the quality of the solutions achieved by the metaheuristics the most current versions (12.5) of CPLEX and CP solvers were used.

4. Computational experiments

The experiments were conducted on the basis of problem instances generated in the same way as proposed by Araujoa et. al. [1]. Ten instances had been generated for the small problem size (I =10, K =2), medium size (I =50, K =10) and the largest size (I =100, K =20). Each problem had a five-day planning horizon with 10 sub-periods each day. Contrary to the experiments performed by Araujoa et. al. only moderate tightness of furnace capacity (i.e. *Cap*=1.0**C*) was considered. Each algorithm (except of CLPEX) was run 20 times for a given instance and the time limit was set to 100, 250 and 450 seconds, depending for the problem size. The results were presented in Table 1 as the relative deviation from the CPLEX solution for the best solution (out of 20) and average solution provided by a particular metaheuristic and, in the last column, the constraint programming solver. A negative number means that the metaheuristic achieved a better result than CPLEX in a given amount of time.

TABLE 1
Relative deviation of metaheuristics from the CPLEX solution

	Small problems (I =10, K =2)						
	GA_{best}	GA_{avg}	TS_{best}	TS_{avg}	DE_{best}	DE_{avg}	CP
avg.	**-0.09**	**-0.01**	0.32	1.77	**-0.03**	0.05	2.14
st.dev.	0.08	0.04	0.46	0.96	0.10	0.10	0.52
	Medium problems (I =50, K =10)						
	GA_{best}	GA_{avg}	TS_{best}	TS_{avg}	DE_{best}	DE_{avg}	CP
avg.	**-0.13**	**-0.02**	0.56	1.01	0.29	0.37	3.07
st.dev.	0.06	0.08	0.26	0.51	0.14	0.4	0.37
	Large problems (I =100, K =20)						
	GA_{best}	GA_{avg}	TS_{best}	TS_{avg}	DE_{best}	DE_{avg}	CP
avg.	**-0.03**	0.15	0.44	0.58	0.43	0.51	2.18
st.dev.	0.03	0.07	0.14	0.15	0.05	0.06	0.15

Best results achieved by our genetic algorithm were better than the ones achieved by CPLEX for all problem sizes. Differential evolution occurred to be competitive to GA and CPLEX only in the case of small problem instances. Tabu search provided usually the worst results of all tested metaheuristics with the except for large problems, when the best results achieved by DE were on average the worst. Most probably, this is due to the fact that TS operates on a single solution, while the remaining algorithms operate on the population of solutions. We also tested simulated annealing algorithm in the version described by Araujoa et. al., but did not managed to achieve feasible solutions (Araujoa et. al., however, used their SA for adjusting a rolling horizon with much smaller solution space than in our experiments). It is worth to observe that constraint programming was significantly weaker even than the worst-performing tabu search metaheuristic.

5. Conclusions

In this paper, the computational intelligence algorithms are proposed for the lot-sizing and scheduling problem in automated foundries. The use of heuristics is motivated by the fact that commercial software like CLPEX can solve production planning problems only if such plant are written as MIP formulation. Although MIP solvers are currently able to handle problem instances of large size, the results provided by commercially available alternative methods (like CP) still cannot be seen as satisfactory. Algorithms that are based on universal

Unauthenticated | 89.73.89.243
Download Date | 12/6/13 3:00 PM

metaheuristics are able to obtain significantly better results than constraint programming and, when properly tuned, they can even compete with the state-of-art CPLEX solver. The genetic algorithm proposed by us can achieve better results than CPLEX, and it can potentially handle more complex problems, which can be expressed in any form (including if-then rules, external functions) that allows to assess the quality of solutions. There are a number of research directions that can be considered as useful extensions of this research, concerning both further metaheuristics development and production planning modelling that more accurately describes the real-world problems.

REFERENCES

[1] S.A. de Araujo, M.N. Arenales, A.R. Clark, Lot sizing and furnace scheduling in small foundries, Comput Oper Res. **35**, 916-932 (2008).

[2] V. Camargo, L. Mattiolli, F. Toledo, A knapsack problem as a tool to solve the production planning problem in small foundries, Comput Oper Res. **39**, 86-92 (2012).

[3] S.K. Gauri, Modeling product-mix planning for batches of melt under multiple objectives in a small scale iron foundry, Prod Eng Res Dev. **3**, 189-196 (2009).

[4] M. Gendreau, J.-Y. Potvin (ed.), Handbook of Metaheuristics. International Series in Operations Research & Management Science **146** (2010).

[5] Y.K. Park, J.-M. Yang, Optimization of mixed casting processes considering discrete ingot sizes, J Mech Sci Technol. **23**, 1899-1910 (2009).

[6] E. dos Santos-Meza, M.O. dos Santos, M.N. Arenales, A lot-sizing problem in an automated foundry, Eur J Oper Res. **139**, 490-500 (2002).

[7] A. Stawowy, J. Duda, Models and algorithms for production planning and scheduling in foundries – current state and development perspectives, Arch Foundry Eng. **12**, 2, 69-74 (2012).

[8] R.F. Teixeira, F. Fernandes, N. Pereira, Binary integer programming formulations for scheduling in market-driven foundries. Comput Ind Eng. **59**, 425-435 (2010).

This article was first presented at the VI International Conference ”DEVELOPMENT TRENDS IN MECHANIZATION OF FOUNDRY PROCESSES”, Inwałd, 5-7.09.2013

Received: 20 January 2013.

Unauthenticated | 89.73.89.243
Download Date | 12/6/13 3:00 PM