RAIRO-Oper. Res. 53 (2019) 1551–1561
https://doi.org/10.1051/ro/2019084

RAIRO Operations Research
www.rairo-ro.org

THE PRODUCTION PLANNING PROBLEM OF ORDERS IN SMALL FOUNDRIES

Maria Gabriela S. Furtado[1,*], Victor C. B. Camargo[2] and Franklina M. B. Toledo[1]

Abstract. The production planning problem in market-driven foundries consists of determining the alloys to be melted and the items to be produced. A production plan is defined based on the foundry book order that compiles customer requests. Each customer order can be composed of items of different types and made from different alloys. In the literature, production planning does not usually consider the customers' orders, *i.e.*, items are independently handled. However, in practical situations, several orders cannot be partially delivered. An order can be delivered only when all of its items have been produced, which often results in delays in meeting the customer demands. Conversely, some orders may be split, which would incur delivering costs. This paper addresses the production planning problem of orders in market-driven foundries, which is considered a gap in the literature. A mathematical model and a relax-and-fix heuristic are proposed to address the problem. An experimental evaluation using synthetic datasets and a real dataset compares the proposed methods with the practical policy planning and a state-of-the-art method that solve the production planning problem of items. The results demonstrate the relevance of the production planning of orders.

Mathematics Subject Classification. 90C11, 68T20, 90C90.

Received October 8, 2016. Accepted September 6, 2019.

1. Introduction

Market-driven foundries are suppliers of different sectors of the economy, such as the automotive, agricultural and oil and gas sectors. The production planning in these foundries involves a large variety of items and alloys, which results in a difficult and challenging problem. The decision maker must determine an operational plan that set outs the alloys to be melted and the items to be produced from them.

Generally, small foundries have only one furnace with limited production capacity. Consequently, the furnace is the production bottleneck. Two points are essential to this problem: (i) delivery of orders on time, *i.e.*, minimization of possible delays in the customer orders, and (ii) improvement of the furnace capacity usage.

Keywords. Market-driven foundry, lot sizing, mixed integer programming, relax-and-fix heuristic.

[1] Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, Caixa Postal 668, São Carlos – SP, CEP 13560-970, Brazil.

[2] Departamento de Engenharia de Produção, Universidade Federal de São Carlos, Rodovia Washington Luís, km 235, São Carlos – SP-310, SP 13565-905, Brazil.

[*]Corresponding author: gabisfurtado@gmail.com

Article published by EDP Sciences

This problem has been tackled in the literature due to its relevance to the real environment of industry and challenge to academic research.

Stawowy and Duda [14] surveyed the state-of-the-art of production planning in foundries. Santos-Meza *et al.* [12], Araujo and Arenales [1], Duda [6] and Duda and Osyczka [7] targeted production planning in large and medium-sized foundries. Araujo *et al.* [2], Tonaki and Toledo [17], Teixeira *et al.* [16], Camargo *et al.* [4] and Duda and Stawowy [8] proposed models and methods to solve the problem in small foundries, which is the question posed here.

Araujo *et al.* [2] designed a mixed integer model to represent the production planning problem in small foundries and solved it using heuristic methods. Tonaki and Toledo [17] decomposed the problem into two sub-problems to be hierarchically solved using Lagrangian heuristics. Camargo *et al.* [4] proposed a genetic algorithm incorporating a knapsack problem to solve the decomposed problem. Teixeira *et al.* [16] studied the balancing and synchronizing production of the molding, pouring and casting processes. The authors presented a binary integer model for the problem and proposed a heuristic based on linear relaxation to solve the problem. Duda and Stawowy [8] presented three methods to solve the problem: a genetic algorithm, a tabu search, and differential evolution. The authors concluded that the genetic algorithm achieved the best results for the instances. All these studies deal with the production planning of foundries focusing on items, that is, when an item is produced, it is immediately considered delivered to the customer.

As highlighted in Camargo [3] and observed in practice, an order can be composed of many items made by different alloys, and in several real cases, the partial delivery of the order is not possible. For example, if items are parts of a final product or a customer stipulates that all of the items of the order must be delivered together, partial deliveries are prohibited. In other situations, partial deliveries are convenient for some customers of integrated supply chains and/or to reduce the inventory at the foundry, even if additional penalties are involved. Although Teixeira *et al.* [16] also indicated the importance of orders, those orders are composed of only one item type or items of the same alloy.

Management of the items in orders involves both production and delivery decisions. Studies integrating production and transportation decisions are reported in the review of Chen [5]. The author observes that most of these problems lie at the strategic or tactical levels. In contrast, Stecke and Zhao [15] proposed mathematical models for sequencing production and transportation of orders. The considered orders have only one product type, and complex manufacturing characteristics do not limit the production capacity. To the best of our knowledge, no study has addressed the influence of the order production sequence in capacity usage at the operational level, especially considering the specificities of market-driven foundries.

This paper proposes a mathematical model for the planning of furnace loads and the items to be produced with minimum delays in their deliveries. Multi-delivery orders are possible, but they incur additional penalties. A relax-and-fix heuristic has been designed to address the new problem. Our proposals were subjected to computational tests with instances based on real datasets.

The paper is organized as follows: Section 2 introduces the model for the production planning problem of orders, Section 3 describes the relax-and-fix method proposed to solve the problem, Section 4 reports the computational experiments conducted, and finally, Section 5 provides the concluding remarks and future research directions.

2. Mathematical model proposed

The basic processes for the production of items in market-driven foundries can be summarized as follows: building the molds, preparing and merging the alloys in the furnace, pouring the metal into the molds and removing any burrs. These foundries usually manufacture a wide range of items made from different types of alloys. Due to such a variety, production planning is a hard manual task, and handmade plans often result in long delays for some orders.

In some cases, multi-deliveries of an order are desirable, but they are limited and may incur additional penalties. In other cases, customers impose only one delivery, mainly because of two reasons: (1) the items of an order are related to the same final product, and therefore, their separate delivery is not possible, or (2) multi-deliveries result in high delivery costs. The delivery costs are high in market-driven foundries particularly because limited human resources are involved in imposing the payment of overtime.

By approaching production planning with these additional features, we aim to minimize the delay to meet the customer orders. The planning horizon is finite and divided into periods and each period is split into small periods (subperiods), that is, periods represent days of the horizon, and subperiods represent furnace loads. The loadings are limited, and only one alloy can be merged per subperiod. Each item is produced by one type of alloy. The customer orders are known *a priori*, and the demand is met at the end of each period. One order might demand various items produced from different alloys, and the order is considered fulfilled when all its items are produced. Furthermore, orders can be partially delivered according to the customer needs or the foundry policies. As proposed by Araujo *et al.* [2], the alloy changeover cost was evaluated to be minimized. However, as also observed in foundry practice, this cost is not a significant objective in the decision-making process.

Further technological constraints faced by the foundry industry must also be respected and were proposed by Araujo and Arenales [1]. Based on such features, we propose the mathematical model for the production planning problem of orders (P3O), the notation for which is given as follows.

Indices and sets	
$k \in \{1,\ldots,K\} = \mathcal{K}$	Types of alloys;
$i \in \{1,\ldots,NP\} = \mathcal{I}$	Orders;
$j \in \{1,\ldots,N\} = \mathcal{N}$	Types of items;
$t \in \{1,\ldots,T\} = \mathcal{T}$	Time periods;
$\eta \in \{1,\ldots,L_T\} = \mathcal{L}$	Subperiods;
$S(i)$	Items that belong to order i;
$A(j)$	Orders that contain item j;
$L(k)$	Items to be produced with alloy k.

Parameters	
NS_t	Number of subperiods in period t, *i.e.*, furnace loads in period t;
$F_t = 1 + \sum_{r=1}^{t-1} \mathrm{NS}_r$	First subperiod in period t ($F_1 = 1$);
$L_t = \sum_{r=1}^{t} \mathrm{NS}_r$	Last subperiod in period t;
bo_{it}	Penalty for backlogging order i in period t;
pe_{it}	Penalty for extra-delivery of order i in period t;
r_{jit}	Benefit of delivering in advance item j of order i in period t;
h_j	Benefit of holding item j in the end of planning horizon (period T);
d_{it}	Takes 1 if the due date of order i is in period t; 0 otherwise;
ρ_j	Gross weight (kg) of item j;
cap	Single furnace loading capacity (kg);
me_i	Maximum number of extra-deliveries of order i;
a_{ji}	Units of item j demanded in order i.

Variables	
BO_{it}	Takes 1 if order i is delayed in period t; 0 otherwise;
E_{it}	Takes 1 if there is an extra-delivery of order i in period t; 0 otherwise;
Γ_{jit}	Auxiliary variable that indicates the quantity of item j of order i delivered in period t in an extra-delivery;
I_{jt}	Number of items j stocked at the end of period t;
XO_{it}	Takes 1 if order i is concluded in period t; 0 otherwise;
$X_{j\eta}$	Number of items j to be produced in subperiod η;
Y_η^k	Takes 1 if the furnace has been used for the production of alloy k in subperiod η; 0 otherwise;
W_{jit}	Amount of items j of order i delivered in period t.

The P3O model reads as follows:

$$\text{(P3O) Minimize} \sum_{i\in\mathcal{I}}\sum_{t\in\mathcal{T}}(\mathrm{bo}_{it}\cdot \mathrm{BO}_{it}+\mathrm{pe}_{it}\cdot E_{it})-\sum_{t\in\mathcal{T}}\sum_{\substack{i\in\mathcal{I}\\ j\in S(i)}} r_{jit}\cdot\Gamma_{jit}-\sum_{j\in\mathcal{N}}h_j\cdot I_{jT} \tag{2.1}$$

subject to:

$$\mathrm{XO}_{it}+\mathrm{BO}_{it}=d_{it}+\mathrm{BO}_{i,t-1} \qquad i\in\mathcal{I};t\in\mathcal{T} \tag{2.2}$$

$$I_{j,t-1}+\sum_{\eta=F_t}^{L_t}X_{j\eta}=I_{jt}+\sum_{i\in A(j)}W_{jit} \qquad j\in\mathcal{N};t\in\mathcal{T} \tag{2.3}$$

$$\sum_{k\in\mathcal{K}}Y_\eta^k\leq 1 \qquad \eta\in\mathcal{L} \tag{2.4}$$

$$\sum_{j\in L(k)}\rho_j\cdot X_{j\eta}\leq \mathrm{cap}\cdot Y_\eta^k \qquad \eta\in\mathcal{L};k\in\mathcal{K} \tag{2.5}$$

$$\sum_{\eta\in\mathcal{L}}X_{j,\eta}\leq\sum_{i\in\mathcal{I}}a_{ji} \qquad j\in\mathcal{N} \tag{2.6}$$

$$W_{jit}\leq a_{ji}\left(\mathrm{XO}_{it}+E_{it}\right) \qquad j\in S\left(i\right);i\in\mathcal{I};t\in\mathcal{T} \tag{2.7}$$

$$\sum_{t\in\mathcal{T}}W_{jit}\leq a_{ji} \qquad j\in\mathcal{N};i\in A\left(j\right) \tag{2.8}$$

$$\mathrm{XO}_{it}+E_{it}\leq 1 \qquad i\in\mathcal{I};t\in\mathcal{T} \tag{2.9}$$

$$\sum_{l=1}^{t}W_{jil}\geq a_{ji}\cdot\mathrm{XO}_{it} \qquad j\in S\left(i\right);i\in\mathcal{I};t\in\mathcal{T} \tag{2.10}$$

$$\sum_{t\in\mathcal{T}}E_{it}\leq\mathrm{me}_i \qquad i\in\mathcal{I} \tag{2.11}$$

$$\Gamma_{jit}\leq\min\left\{W_{jit};a_{ij}\left(1-\mathrm{XO}_{it}\right)\right\} \qquad j\in S\left(i\right);i\in\mathcal{I};t\in\mathcal{T} \tag{2.12}$$

$$\mathrm{BO}_{it},\mathrm{XO}_{it},E_{it}\in\{0,1\} \qquad i\in\mathcal{I};t\in\mathcal{T} \tag{2.13}$$

$$Y_\eta^k\in\{0,1\} \qquad \eta\in\mathcal{L};k\in\mathcal{K} \tag{2.14}$$

$$W_{jit},X_{j\eta}\in\mathbb{Z}^+ \qquad j\in\mathcal{N};i\in\mathcal{I};t\in\mathcal{T};\eta\in\mathcal{L} \tag{2.15}$$

$$\Gamma_{jit},I_{jt}\geq 0 \qquad j\in\mathcal{N};i\in\mathcal{I};t\in\mathcal{T} \tag{2.16}$$

The first part of the objective function (2.1) aims to minimize the sum of backlogged orders and extra-delivery penalties. The second indicates the benefits obtained by each item delivered in advance. These benefices are computed in case multi-deliveries are allowed. Furthermore, incentives are considered for items stored in the last period, so that the furnace usage can be optimized.

Equations (2.2) define when each order is completed. In case, the order i is not completed the variable BO_{iT} is equal to one. Equations (2.3) establish that the sum of produced items and current inventory is enough for delivery and inventory at the end of period t. Note that the inventory of an order is represented by the inventory of its items, *i.e.*, if all items of the order are produced but at least one is not delivered, the order is still open. In this model, an order is considered concluded just when all its items are delivered.

Constraints (2.4)–(2.6) define the features for production and furnace usage. Constraints (2.4) impose that, at most, only one alloy is melted in each subperiod. Constraints (2.5) maintain the production within the furnace capacity. The production of items is limited to the amount demanded by the constraints (2.6).

Constraints (2.7)–(2.12) define the concluded orders, delivered items and extra deliveries. Constraints (2.7) determine that deliveries only occur in case the order is concluded or it is an extra delivery. Note that delivering the produced items in advance can be used to minimizing the total cost, the total of delivered items is limited to the amount demanded by the constraints (2.8). Moreover, extra deliveries occur only if the order is still open in that period; see (2.9). Constraints (2.10) impose that an order can be concluded only if all the items have been delivered. Constraints (2.11) set an upper bound on the number of extra deliveries. Constraints (2.12) limit the number of items delivered in advance (this limit is determined by the entire order). Observe that an extra delivery does not occur if an order is concluded. Constraints (2.16) and (2.15) define the variable domains.

Remark 2.1. The following valid inequalities are added to P3O to avoid symmetric solutions:

$$\sum_{k\in\mathcal{K}} Y_{\eta}^{k} \geq \sum_{k\in\mathcal{K}} Y_{\eta+1}^{k} \qquad t\in\mathcal{T};\eta\in N(t)\backslash L_t, \tag{2.17}$$

where $N(t) = \{F_t,\ldots,L_t\}$ is the set of subperiods in period t.

Remark 2.2. If customers or the foundry's policies require only one delivery for the orders, me_i is set to zero for $i\in\mathcal{I}$.

3. Relax-and-fix heuristic

A relax-and-fix heuristic to solve P3O is proposed in this section. The method is simple in terms of implementation and has shown good performance in terms of solving lot-sizing problems [9–11, 13, 18].

The basic idea is to divide a set Q of integer variables into M disjoint subsets represented by Q^m, $m = 1,\ldots,M$, such that $Q = Q^1\cup\ldots\cup Q^M$. Besides that, the intersection of any two sets Q^i and Q^j is the empty set. In each iteration, the variables are considered as fixed-value, integer variables or linear relaxed variables. In the first iteration of the heuristic, the variables of subset Q^1 are integer variables, and those of $Q\backslash Q^1$ are relaxed variables. In the second iteration, variables of Q^1 are fixed, to the values obtained in the first iteration. Subset Q^2 takes up integer variables, and $Q\backslash\left(Q^1\cup Q^2\right)$ remain as relax variables. Iteratively, the solution to this problem provides values so that these variables are fixed in the next iteration. The execution is stopped when all variables in Q are integer values.

In our adaptation, the relax-and-fix heuristic manages binary variables Y_{η}^{k} and integer variables $X_{j\eta}$. Recall that Y_{η}^{k} indicates that alloy k is to be merged into subperiod η and that $X_{j\eta}$ represents the amount of item j to be produced in subperiod η. In other words, Q contains Y_{η}^{k} and $X_{j\eta}$ and is divided into T subsets representing the periods of the planning horizon (each subset is related to a period). The following problem is solved at each iteration of the relax-and-fix heuristic. Let $\left\{\overline{Y}_{\eta}^{k},\overline{X}_{j\eta}\right\}$ be the incumbent solution of the relax-and-fix MIP heuristic at iteration $t-1$ ($\mathrm{MIP_{RF}}(t-1)$). The problem reads as follows:

$$\mathrm{MIP_{RF}}(\mathrm{t}) = \text{Minimize} \sum_{i\in\mathcal{I}}\sum_{t\in\mathcal{T}}\left(\mathrm{bo}_{it}\cdot \mathrm{BO}_{it} + \mathrm{pe}_{it}\cdot E_{it}\right) - \sum_{t\in\mathcal{T}}\sum_{\substack{i\in\mathcal{I}\\ j\in S(i)}} r_{jit}\cdot\Gamma_{jit} - \sum_{j\in\mathcal{N}} h_{jT}\cdot I_{jT} \tag{2.1}$$

subject to

(2.3)–(2.13)

$$Y_{\eta}^{k} = \overline{Y}_{\eta}^{k},\ X_{j\eta} = \overline{X}_{j\eta} \qquad Y_{\eta}^{k},\ X_{j\eta}\in\bigcup_{m=1}^{t-1} Q^m, t\in\mathcal{T}\backslash\{1\} \tag{3.1}$$

$$Y_{\eta}^{k}\in\{0,1\},\ X_{j\eta}\in\mathbb{Z}^{+} \qquad Y_{\eta}^{k},\ X_{j\eta}\in Q^t \tag{3.2}$$

$$0\leq Y_{\eta}^{k}\leq 1,\ X_{j\eta}\geq 0 \qquad Y_{\eta}^{k},\ X_{j\eta}\in Q\backslash\bigcup_{m=1}^{t} Q^m. \tag{3.3}$$

In each iteration t, a mixed integer problem ($\mathrm{MIP_{RF}}(\mathrm{t})$) is solved. Thereby, the number of subsets defines the number of iterations of our relax-and-fix adaptation (T iterations). Once backlogging is allowed in the studied problem, every subproblem has a feasible solution.

Algorithm 1 describes the relax-and-fix heuristic adapted from Ferreira *et al.* [10].

Algorithm 1: Relax-and-fix heuristic.

Define set Q and its subsets Q^t;
Start the incumbent solution as empty;
$t = 0$;
repeat
  $t = t + 1$;
  Generate $\mathrm{MIP}_{\mathrm{RF}}\,(t)$ relaxing all the variables, then:
    Define the variables of Q^t to be integer;
    Fix the variables of $\bigcup_{m=1}^{t-1} Q^m$ to the values previously found;
    Insert the incumbent solution of $\mathrm{MIP}_{\mathrm{RF}}\,(t-1)$ as an initial solution;
  Solve $\mathrm{MIP}_{\mathrm{RF}}(t)$ until the optimal solution has been obtained, or the maximum running time has been reached;
  Update the incumbent solution;
until $t = T$;

4. Computational results

The formulation P3O is innovative, as it addresses the production planning of orders in foundries. An essential research question is related to the relevance of this problem in comparison to the production planning of items. In other words, does the production planning of items implicitly deal with the order composition? In order to answer this question, in the first phase of the computational tests, we compared the number of concluded orders given by P3O when extra deliveries were not allowed (*i.e.*, the order must be delivered only if all the items have been produced ($\mathrm{me}_i = 0$)) against a state-of-the-art method proposed for solving the production planning of items (GAKP – [4]). In GAKP, an individual represents the production sequence of alloys. For each, alloy, a knapsack problem was solved to find the best allocation of items in the furnace. We also compared P3O with the practical results. The practical results were simulated on the software currently used by the foundry. In the second phase of the computational tests, we allowed one extra delivery per order ($\mathrm{me}_{it} = 1$). The aim was to test the P3O application on integrated supply chains or when multi-deliveries are allowed (in practice it can happen).

The experiments were performed using CPLEX 12.5 software (set with one thread and its other parameters as default) and on an Intel Xeon 2.0 GHz computer with 64 GB of RAM and an Ubuntu 12.04 operating system. The runtime for each instance was limited to one hour (common limit used by the literature of production planning).

4.1. Instances description

The computational tests were run based on a real order book from a foundry located in the state of São Paulo, Brazil, presented in Araujo *et al.* [2]. The order book contains 19 different alloys and 383 items to be produced. The number of periods (T) is 5 days. The foundry has one furnace of 360 kg capacity that can be used at most 10 times per day, *i.e.*, the number of subperiods is equal to 10, totalizing 50 subperiods in the planning horizon. Based on this order book, Tonaki and Toledo [17] proposed 10 order books that have cases of 80%, 90% and 100% of delayed items and planning horizons of 3 and 5 periods.

For these 11 order books (one from [2] and 10 from [17]), the items are not associated with orders; therefore, the orders were generated based on the due dates of the items. Three classes of instances were proposed for each order book. In the first, with a minimum number of orders (called $M1$), all items of the same due date belong to the same order. In the second class, called $M2$, the items are allocated to random orders. In the third class, with the maximum number of orders ($M3$), each item is an order. Note that the class of the maximum number of orders ($M3$) resembles the item approach, in which each type of item is considered an order. $M1$ and $M2$ classes include orders with different types of items. In total, there are 33 instances.

For each instance, the inventory and delay penalties are based on the penalties defined in Araujo *et al.* [2]. The new parameters is given as follows:

- The weight of the heaviest order is defined by $\max O = \max_{i\in\mathcal{I}}\left\{\sum_{j\in S(i)} a_{ji}\rho_j\right\}$.
- The cost for delaying order i in period t is based on the sum of its items weights, *i.e.*, $\text{bo}_{it} = \sum_{t=1}^{T}\sum_{\substack{i\in\mathcal{I}\\ j\in S(i)}} \left(a_{ji}\rho_j\left(\alpha_i+1+t\right)\right)/\max O$, where α_i is the initial delay of order i (in periods). In case the order is not delayed, bo_{it} is a large number.
- The benefit of delivering in advance item j of order i in period t is $r_{jit} = \text{Rcd}\sum_{t=1}^{T}\rho_j\left(\alpha_i+1+t\right)/\max O$, where Rcd indicates the reduction in the backlogging cost and is defined by the decision maker. For the tests performed, Rcd was equal to 0.5.
- The benefit of holding item j at the end of planning horizon is $h_j = \rho_j/\max O$.
- After empirical analysis of several values for extra deliveries, their penalties (pe_{it}) were set to 0.2.

4.2. Phase I – Experiments with P3O model

The results achieved by P3O, GAKP and a simulated practical approach for the three classes of instances previously defined are shown in Table 1. In Table 1, the first column (Instances) identifies each instance with its total number of orders. For example, 1/43 is the instance number 1 with 43 orders. Column P3O shows the Gap and number of orders concluded at the end of the planning horizon. The Gap is as follows:

$$\text{Gap} = 100\cdot\frac{\text{UB}-\text{LB}}{\text{LB}}$$

where LB and UB are the lower and upper bounds, respectively, obtained when the time limit is reached. LB is the value of the relaxed formulation and the UB the best integer solution. In order to fairly compare the results, the same denominator (LB) is used in each instance to calculate the Gap for the mathematical model and the proposed heuristic. Column *Orders* shows the number of concluded orders. Column *Kg* indicates the amount of waste by furnace under usage. The absolute value of the waste is the difference between the total furnace capacity used and the volume of alloy produced. The table also provides the percentage of waste. This parameter indicates the real furnace capacity used. Table 1 presents the same columns for both GAKP and the practical approach.

Initially, one should note that the approaches work under different objectives. The objective of P3O formulation is to maximize the number of concluded orders, on the other hand, the objective of the GAKP is to maximize the item production. The practical approach provides a compromise between concluding the orders and the optimize usage of the furnace. CPLEX spent one hour to solve P3O for each instance, while GAKP heuristic and practical approach spent no more than one minute to solve.

As expected, on average, P3O can conclude more orders than can GAKP and the practical approach. Indeed, from 33 instances, GAKP found a better solution only for Instance 6 of the $M3$ class. Its seems natural that GAKP performs well in this class, considering that each order corresponds to an item. In addition, P3O concluded all the orders for 8 of the 33 instances, whereas GAKP concluded all the orders for only one instance. P3O also found better solutions or the same solutions for all instances compared with the results from the software used in practice.

By comparing the furnace usage, results from P3O, GAKP and the practical approach indicate 7.9%, 6.8% and 7.8% of waste (on average), respectively. We remark that the production plan given by GAKP is the same for the three classes because GAKP ignores the order feature.

Based on the results, we can conclude that the production planning of orders in small foundries was not implicitly covered by the production planning of items. Furthermore, the furnace under-usage caused by P3O plans is similar to the practice plans. The average P3O gaps suggest the difficulty of proving that the best solutions obtained for these instances are optimal.

TABLE 1. Computational results, Phase I: comparison of P3O, GAKP and the practical approach.

Instances	Gap (%)	P3O Orders	P3O Kg	P3O Waste (%)	GAKP Orders	GAKP Kg	GAKP Waste (%)	Practice Orders	Practice Kg	Practice Waste (%)
Minimum number of orders										
1/43	1.0	42	114	1.1	40	104	1.0	42	528	4.9
2/43	0.8	43	536	3.7	41	644	4.5	42	604	4.2
3/46	1.1	43	6	0.1	41	49	0.5	42	534	4.9
4/48	8.9	43	133	0.7	43	146	0.8	43	594	3.3
5/47	5.5	40	1106	10.2	35	839	7.8	39	1646	15.2
6/47	26.2	46	1656	9.2	44	1523	8.5	46	1889	10.5
7/47	7.0	42	1401	13.0	38	818	7.6	39	1861	17.7
8/47	11.9	46	2787	15.5	41	2395	13.6	44	2043	11.8
9/47	8.7	42	1402	13.0	38	842	7.8	39	1652	15.3
10/47	10.4	47	4269	23.7	45	2505	15.8	46	1937	12.5
11/52	15.4	46	1425	7.9	39	1235	6.9	46	1871	10.4
Average	8.8	43.6	1349	8.9	40.4	1009	6.8	42.5	1378	10.1
Medium number of orders										
1/104	0.3	103	387	3.6	96	104	1.0	98	165	1.5
2/104	0.7	104	536	3.7	100	644	4.5	98	251	1.8
3/137	0.7	112	13	0.1	101	49	0.5	100	40	0.4
4/170	3.7	132	133	0.7	120	146	0.8	121	163	0.9
5/136	1.7	119	1037	9.6	97	839	7.8	104	1200	11.1
6/136	11.4	131	1836	10.2	118	1523	8.5	124	1612	9.0
7/135	2.3	115	814	7.5	110	818	7.6	109	1197	11.1
8/135	8.2	135	2559	14.2	127	2395	13.6	124	1678	9.9
9/135	3.1	117	814	7.5	101	842	7.8	103	1202	11.1
10/135	14.7	135	3549	20.5	130	2505	15.8	133	1548	10.2
11/217	22.7	156	1489	8.3	136	1235	6.9	120	1294	7.2
Average	6.3	123.5	1197	7.8	112.3	1009	6.8	112.2	941	6.7
Maximum number of orders										
1/165	0.6	164	416	3.9	158	104	1.0	159	170	1.6
2/165	0.6	165	536	3.7	165	644	4.5	163	253	1.8
3/228	0.4	174	12	0.1	156	49	0.5	159	66	0.6
4/293	14.2	224	139	0.8	193	146	0.8	198	159	0.9
5/225	1.4	195	842	7.8	178	839	7.8	178	1202	11.1
6/225	9.4	216	1399	7.8	218	1523	8.5	216	1611	9.0
7/224	2.1	197	847	7.8	172	818	7.6	178	1202	11.1
8/224	6.9	224	2559	14.2	198	2395	13.6	217	1679	9.9
9/224	2.0	195	841	7.8	167	842	7.8	178	1202	11.1
10/224	8.9	224	2469	15.2	223	2505	15.8	218	1525	10.1
11/383	17.3	257	1545	8.6	235	1235	6.9	227	1257	7.0
Average	5.8	203.2	1055	7.1	187.5	1009	6.8	190.1	939	6.7
Total average	7.0	123.5	1200	7.9	113.5	1009	6.8	114.9	1085	7.9

4.3. Phase II – Experiments with the relax-and-fix heuristic

Table 3 presents tests comparing model P3O and the relax-and-fix heuristic for the same instance set of Phase I, except $\text{me}_{it} = 1$, for all orders. The runtime was limited to 1 h for both P3O and the heuristic. More specifically, the time limit for each relax-and-fix iteration was defined based on the number of periods, as shown in Table 2. Note that the third column indicates a book order with only three periods, then the symbol – indicates that there are no 4 and 5 periods. The heuristic execution time depends on the iteration and instance type. For example, in the $M1$ class, most of the tests in the first iteration had stopped by the time limit. In

TABLE 2. Time limit for running each relax-and-fix iteration.

Iterations	Book order with 5 periods	Book order with 3 periods
1	1800	2400
2	900	900
3	420	300
4	240	–
5	240	–
Total	3600	3600

TABLE 3. Computational results, Phase II: comparison of the quality of solutions between P3O and relax-and-fix heuristic.

Instances	P3O				Heuristic			
	Gap (%)	Orders	Kg	Waste (%)	Gap (%)	Orders	Kg	Waste (%)
Minimum number of orders								
1/43	1.7	42	114	1.1	1.3	42	115	1.1
2/43	7.4	43	536	3.7	5.2	43	536	3.0
3/46	4.4	43	41	0.4	2.6	42	37	0.3
4/48	81.9	44	301	1.7	34.2	45	131	0.7
5/47	10.4	42	1407	13.0	9.9	41	1103	10.2
6/47	56.7	46	1689	9.4	45.4	46	1928	10.7
7/47	11.2	41	1269	11.7	11.2	41	1329	12.3
8/47	38.1	47	2559	14.2	38.9	47	2558	14.2
9/47	12.6	42	1401	13.0	12.2	41	1329	12.3
10/47	43.4	47	3549	20.5	34.2	47	2468	13.7
11/52	68.4	44	1592	8.8	63.3	43	1417	7.9
Average	30.6	43.7	1314	8.9	23.5	43.5	1177	7.9
Medium number of orders								
1/104	1.1	102	118	1.1	1.0	100	116	1.1
2/104	4.4	104	536	3.7	4.0	104	1976	11.0
3/137	1.8	110	17	0.2	1.5	110	9	0.1
4/170	32.7	133	147	0.8	31.3	125	131	0.7
5/136	5.1	121	1133	10.5	5.6	118	858	7.9
6/136	37.1	135	2181	12.1	34.2	129	1820	10.1
7/135	6.3	116	1272	11.8	6.2	119	940	8.7
8/135	32.9	135	2559	14.2	25.2	134	2683	14.9
9/135	3.5	118	813	7.5	5.3	115	805	7.4
10/135	24.6	135	3189	18.8	23.9	135	2468	13.7
11/217	37.8	150	1319	7.3	31.5	162	1442	8.0
Average	17.0	123.5	1208	8.0	15.4	122.8	1204	7.6
Maximum number of orders								
1/165	1.2	160	145	1.3	1.0	160	143	1.3
2/165	4.9	165	536	3.7	5.0	165	1256	7.0
3/228	1.5	173	13	0.1	1.2	173	12	0.1
4/293	86.2	235	455	2.5	12.0	205	133	0.7
5/225	3.4	200	850	7.9	3.8	199	839	7.8
6/225	19.6	220	1412	7.8	16.0	211	1377	7.6
7/224	3.4	201	843	7.8	3.6	194	840	7.8
8/224	12.8	224	2559	14.2	12.9	224	2558	14.2
9/224	3.4	196	844	7.8	3.1	192	841	7.8
10/224	15.9	224	2469	15.2	13.3	224	3188	17.7
11/383	43.5	267	1376	7.6	31.2	220	837	4.6
Average	17.8	205.9	1046	6.9	9.4	197.0	1093	7.0
Total average	21.8	124.4	1189	7.9	16.1	121.1	1158	7.5

the other iterations, the solver could easily find the optimal solution. For the $M3$ class, only the final iterations stopped by reaching the optimal solution.

The results in Table 3 are divided into four columns for each approach: Gap, number of concluded orders, waste generated by the furnace under-usage along the planning horizon and percentage of waste. The optimality gaps of the heuristic solutions were calculated by the lower bound (LB – obtained when the time limit is reached) of the P3O model.

The comparison of P3O and the heuristic shows, in general, that the relax-and-fix heuristic found better solutions (lower gaps) than P3O. The number of concluded orders given by the production plans can be considered similar for both approaches for the instance classes that better represent the order composition ($M1$ and $M2$). However, the waste of alloy in the furnace is significantly smaller in the solutions generated by the relax-and-fix heuristic. Note that this feature is an important advantage for the practical approach because it enables a better use of the furnace. Furthermore, the results also show that when extra deliveries are allowed, it is difficult to prove the optimality of instances evaluated (comparing Gaps of the P3O model in Tables 1 and 3).

5. Conclusions and future research

This study has addressed the production planning problem of customer orders in small foundries. An order may be composed of different items made by different alloys and is considered concluded only when all of its items have been produced and delivered. Some customers require that all the items of their orders must be produced before delivery, and others allow some partial deliveries with additional penalties. Although these features are taken into account by the software used in practice, managers do not describe them as part of the decision-making process during the production planning.

The mathematical model proposed (P3O) represents the production planning of orders in a small foundry where the number of extra deliveries is controlled by the decision maker. Based on this model, we have proposed a relax-and-fix heuristic to deal with harder problems.

The computational results of our approach overcame those of the practical approach and the state-of-the-art approach to the production planning of items. To facilitate the application of our proposal by small foundries, future research should consider the development of new solution methods independent of commercial software. Furthermore, our approach can be useful and effective in other contexts and industrial environments.

Acknowledgements. The authors acknowledge FAPESP (2009/11877-5, 2016/01860-1, 2012/20278-0, 2013/07375-0), FAPEMIG (APQ-00398-14) and CNPq (308761/2018-9, 447421/2014-0) for the financial support and Viviane Tonaki for simulating the practical results. They are also grateful to anonymous referees, the Optimization Laboratory of ICMC/USP (LOt) and Angela C. P. Giampedro from Centro Cultural/USP.

References

[1] S.A. Araujo and M.N. Arenales, Dimensionamento de lotes e programação do forno numa fundição automatizada de porte médio. *Pesquisa Oper.* **23** (2003) 403–420.

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

[3] V.C.B. Camargo, *Um algoritmo evolutivo para o problema de dimensionamento de lotes em fundies de mercado*. Master's thesis, Instituto de Ciências Matemáticas e de Computação – USP (2009).

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

[5] Z.L. Chen, Integrated production and outbound distribution scheduling: review and extensions. *Oper. Res.* **58** (2010) 130–148.

[6] J. Duda, Lot-sizing in a foundry using genetic algorithm and repair functions. In: *EvoCOP. Lecture Notes in Computer Science*. Springer 2005 101–111.

[7] J. Duda and A. Osyczka, Multiple criteria lot-sizing in a foundry using evolutionary algorithms. In: Evolutionary Multi-Criterion Optimization. *Lecture Notes in Computer Science*. Springer Berlin/Heidelberg (2005) 651–663.

[8] J. Duda and A. Stawowy, Optimization methods for lot-sizing problem in an automated foundry. *Arch. Metall. Mater.* **58** (2013) 863–866.

[9] A. Federgruen, J. Meissner and M. Tzur, Progressive interval heuristics for multi-item capacitated lot sizing problem. *Oper. Res.* **55** (2007) 490–502.

[10] D. Ferreira, R. Morabito and S. Rangel, Solution approaches for the soft drink integrated production lot sizing and scheduling problem. *Eur. J. Oper. Res.* **196** (2009) 697–706.

[11] C. Mercé and G. Fontan, MIP-based heuristics for capacitated lotsizing problems. *Int. J. Prod. Econ.* **85** (2003) 97–111.

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

[13] H. Stadtler, Multilevel lot sizing with setup times and multiple constrained resources: internally rolling schedules with lot-sizing windows. *Oper. Res.* **51** (2003) 487–502.

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

[15] K.E. Stecke and X. Zhao, Production and transportation integration for a make-to-order manufacturing company with a commit-to-delivery business mode. *Manuf. Ser. Oper. Manage.* **9** (2007) 206–224.

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

[17] V.S. Tonaki and F.M.B. Toledo, An approach to solve the lot-sizing and scheduling problem in market-driven foundries. *J. Oper. Res. Soc.* **61** (2010) 108–114.

[18] E.A.V. Toso, R. Morabito and A. Clark, Lot sizing and sequencing optimization at an animal-feed plan. *Comput. Ind. Eng.* **57** (2009) 813–821.